BỘ GIÁO DỤC VÀ ĐÀO TẠO

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHÓA D20

Báo Cáo Nhập Môn Trí Tuệ Nhân Tạo

Đề tài

Nhận diện biển số xe bằng phương pháp học sâu

Lớp : CNPM2 - 08

Giảng viên : Trần Đình Quế

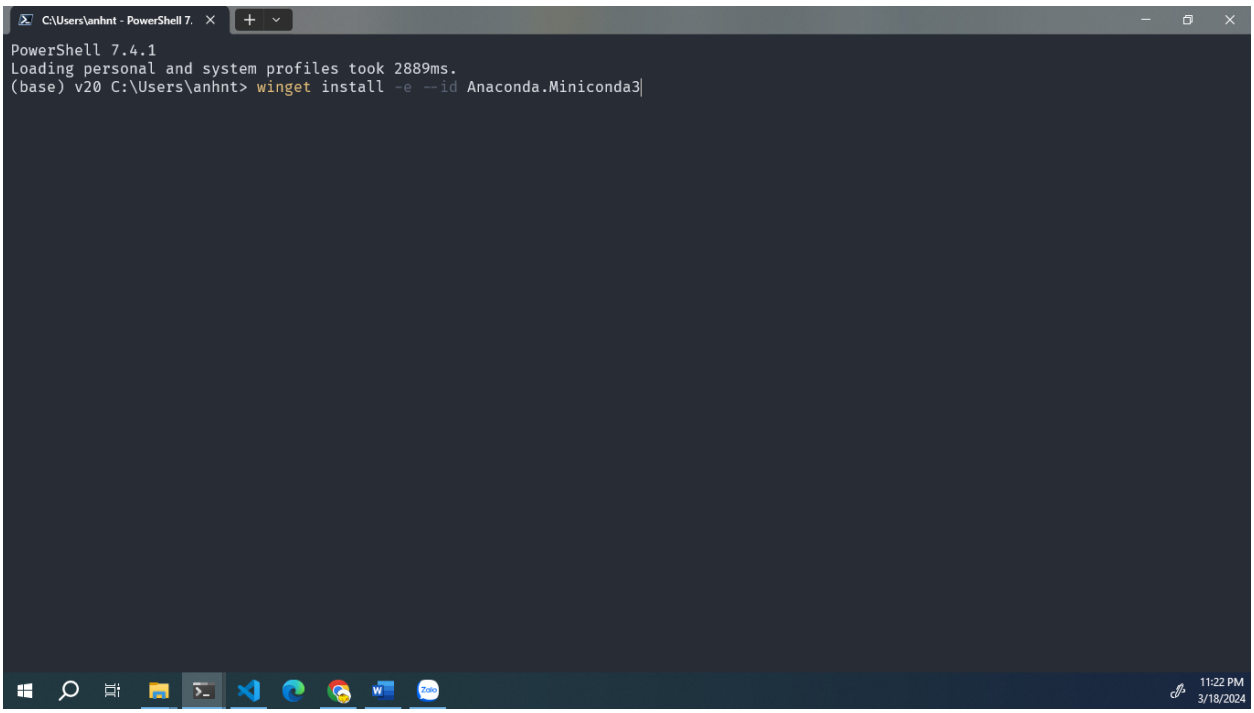
Sinh viên : Nguyễn Tiến Anh

Hà Nội , 2023

I. Tạo 10 services

Tải miniconda để quản lý môi trường và các package cài về của python

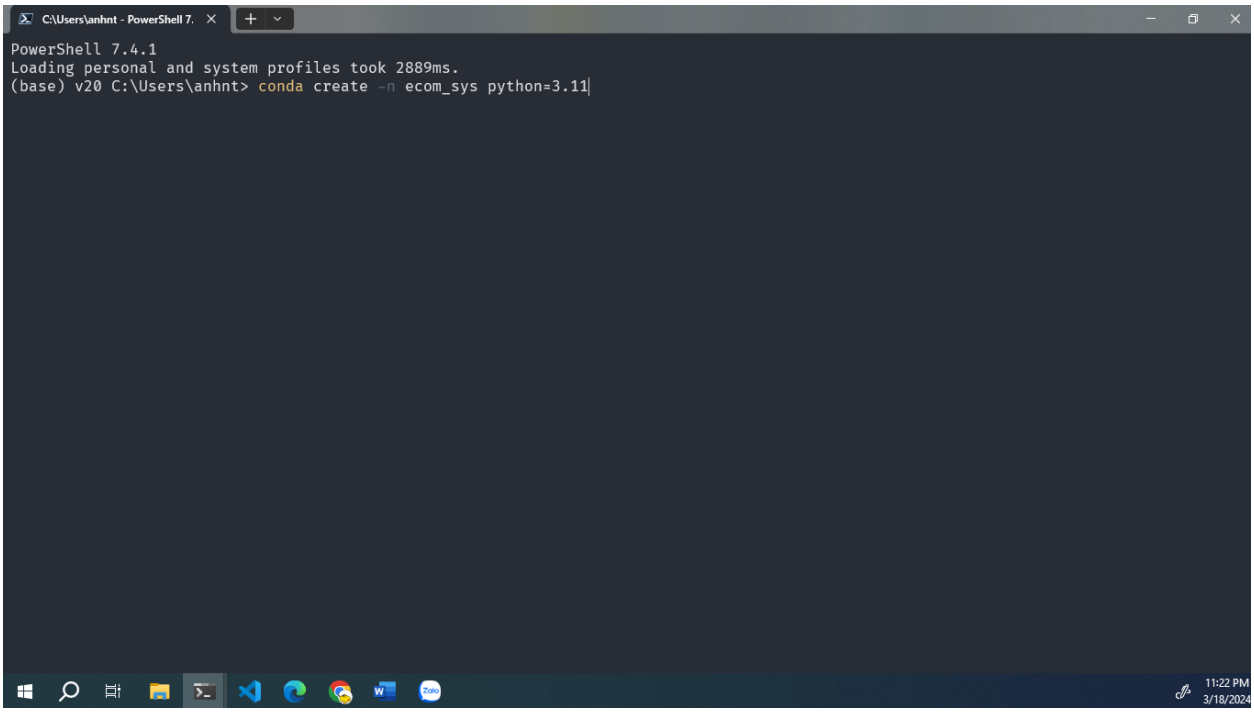
winget install -e --id Anaconda.Miniconda3



Cài đặt miniconda theo hướng dẫn. Sau khi cài đặt xong, thêm path của miniconda vào environment variable của Windows, sử dụng dòng lệnh

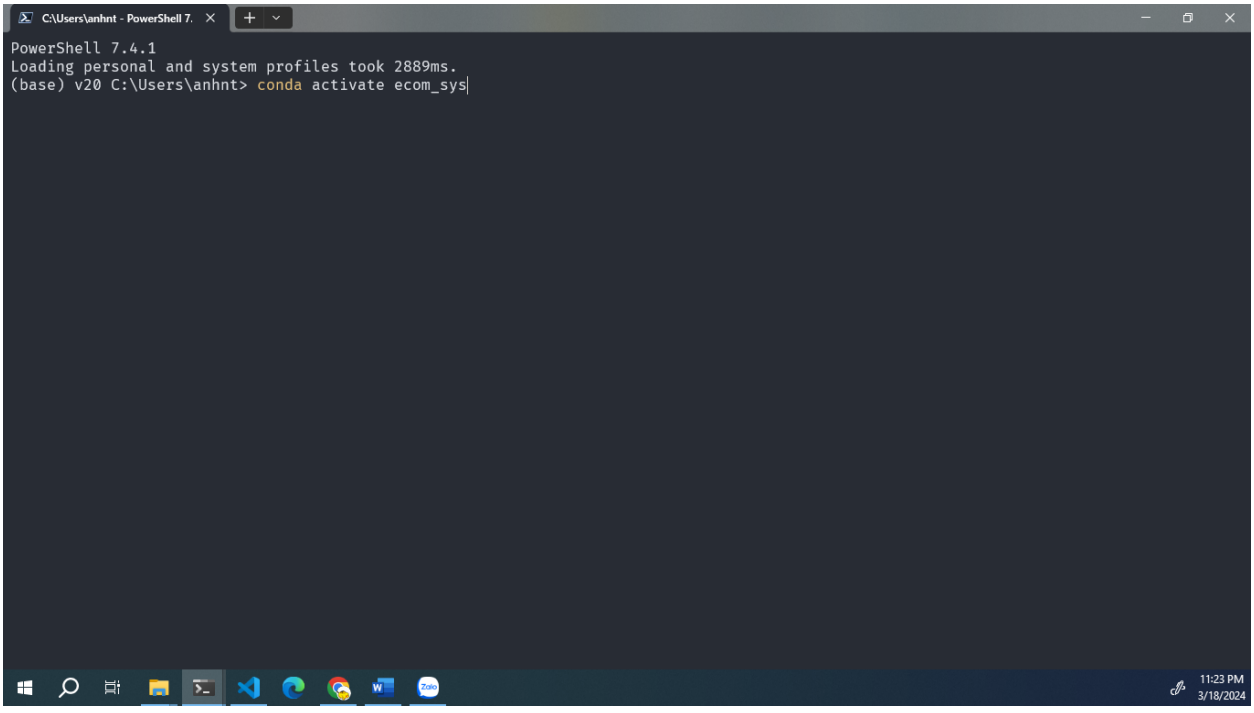
conda create -n ecom_sys python=3.11

để tạo một môi trường quản lý phiên bản python (ở đây là python 3.11).



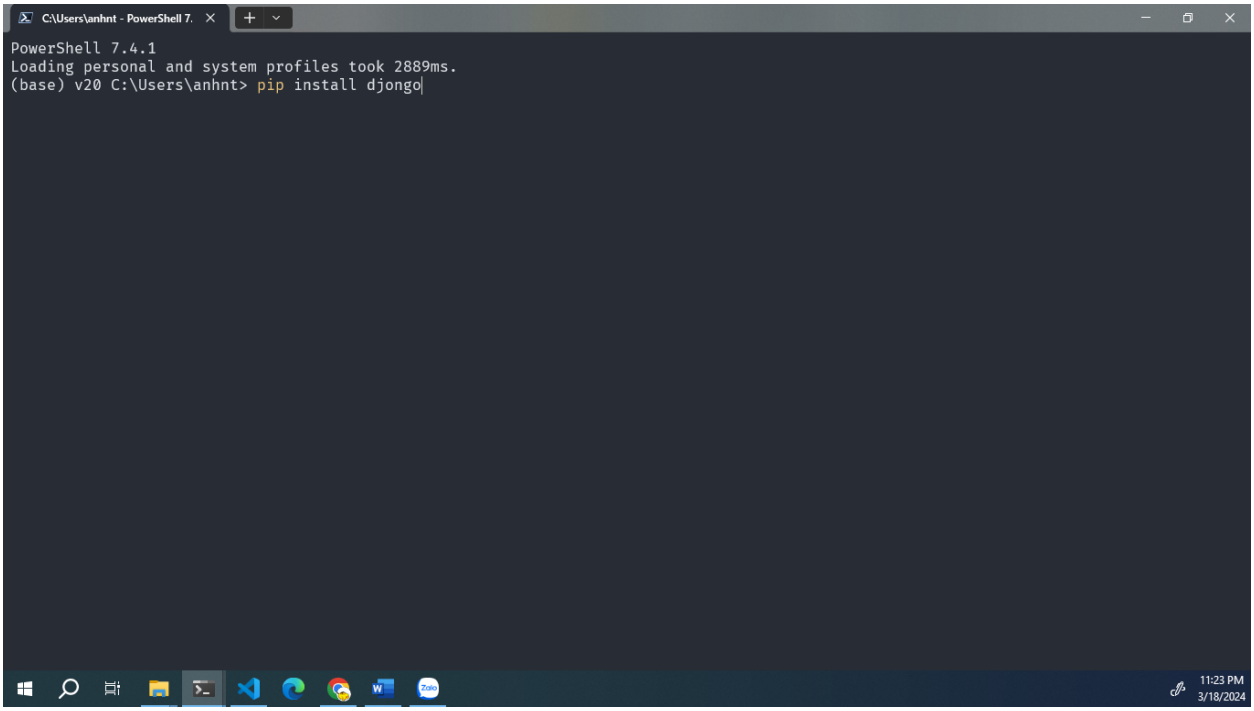
Chuyển môi trường sang ecom_sys bằng dòng lệnh

conda activate ecom_sys



Cài đặt djongo bằng lệnh pip install sau đây

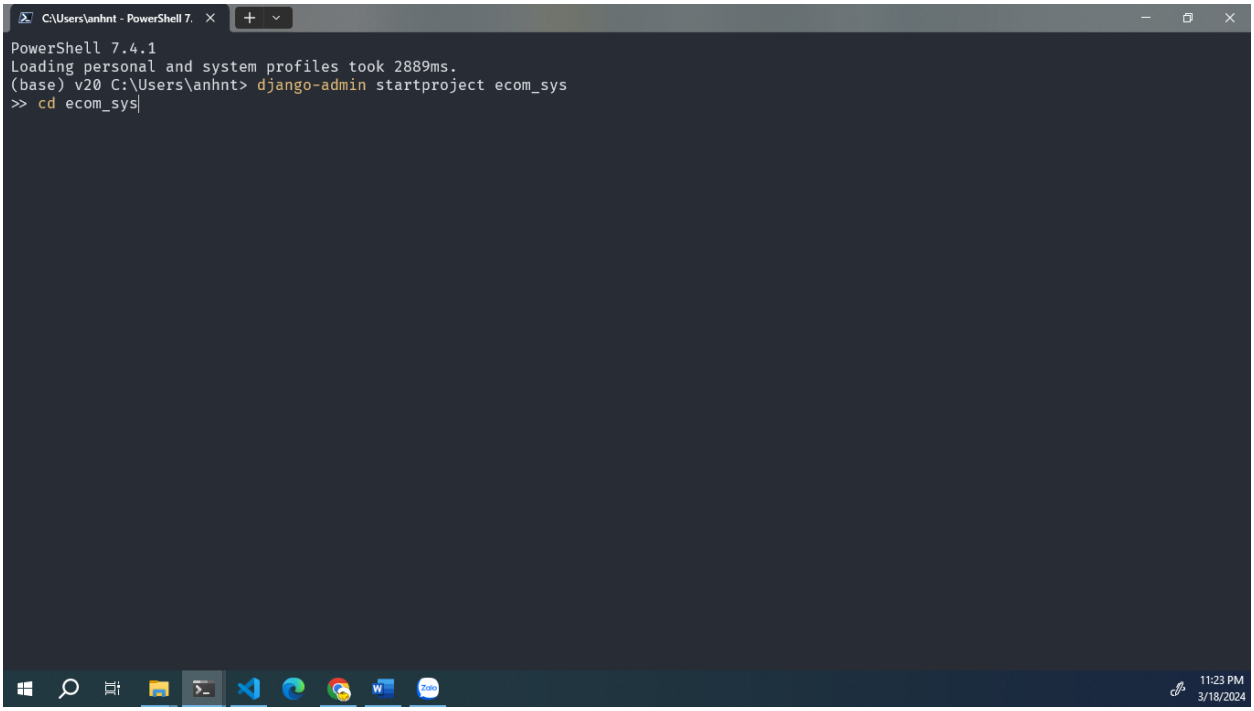
pip install djongo



Django được cài đặt bởi djongo sẽ phù hợp nhất cho việc phát triển ứng dụng Django có database là MongoDB.

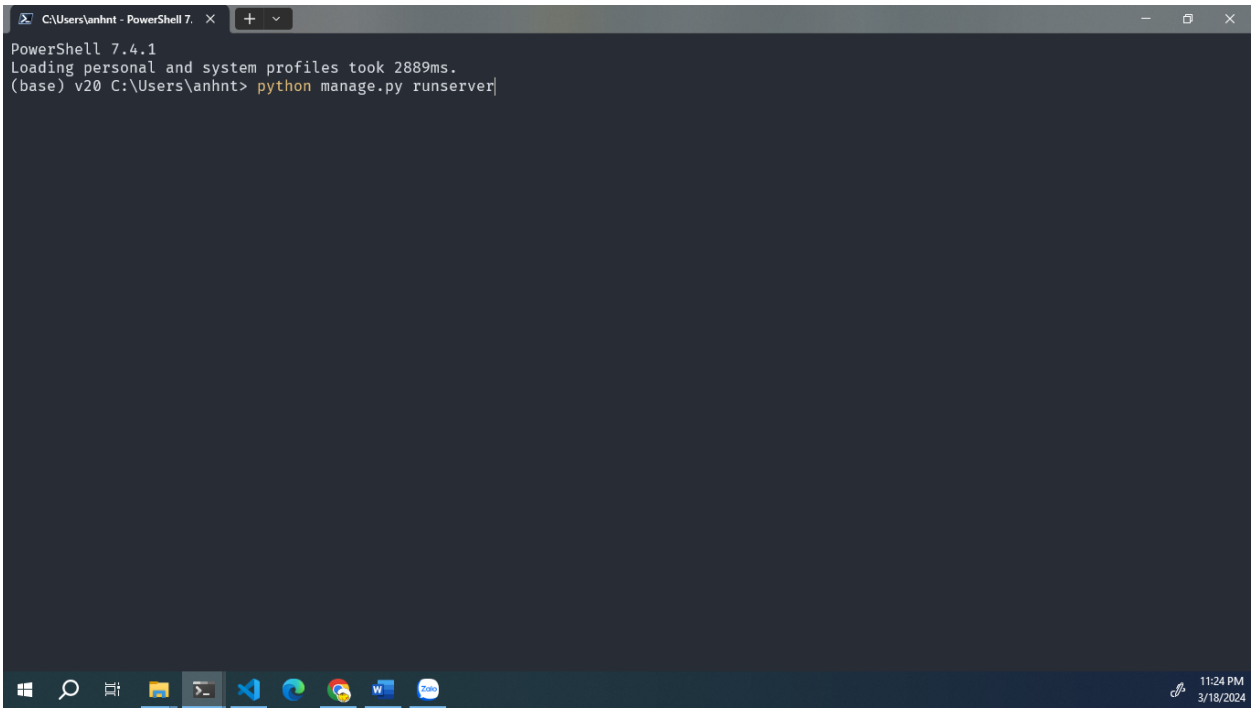
Init project mới bằng lệnh

django-admin startproject ecom_sys
cd ecom_sys



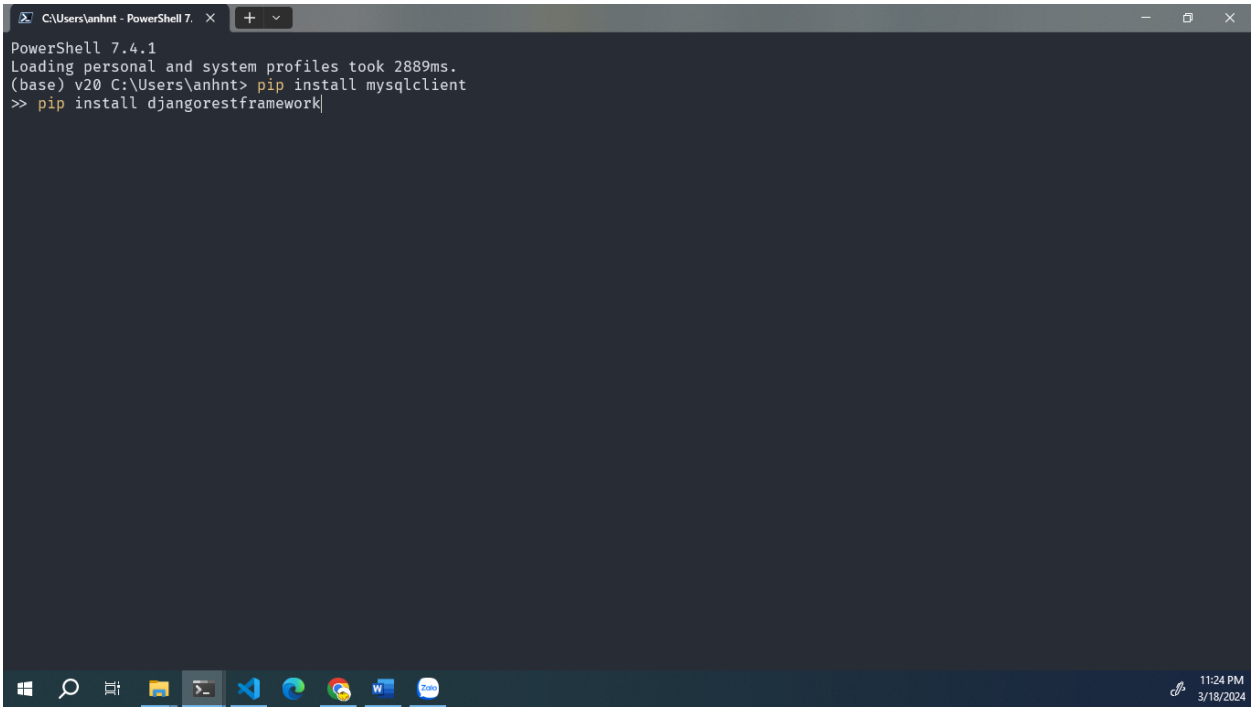
Chạy project lần đầu bằng lệnh

python manage.py runserver

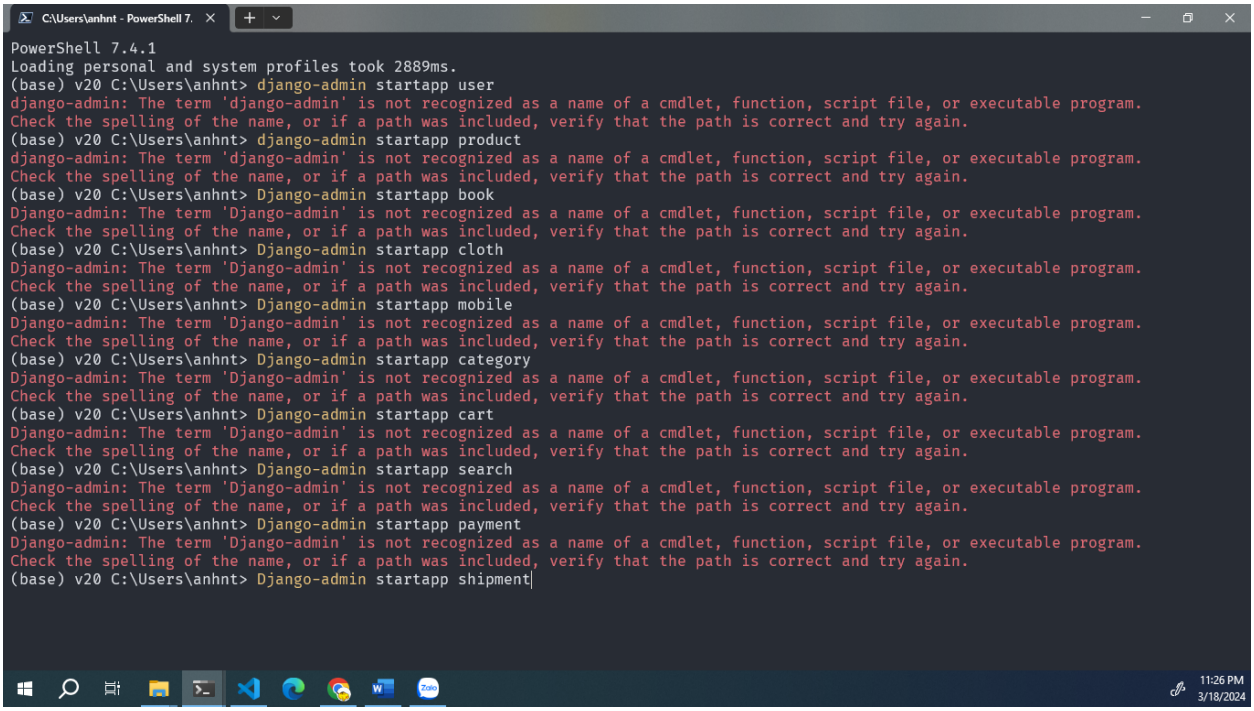


- Cài đặt các package sau đây:
  - mysqlclient: pip install mysqlclient

- Django rest framework: pip install djangorestframework

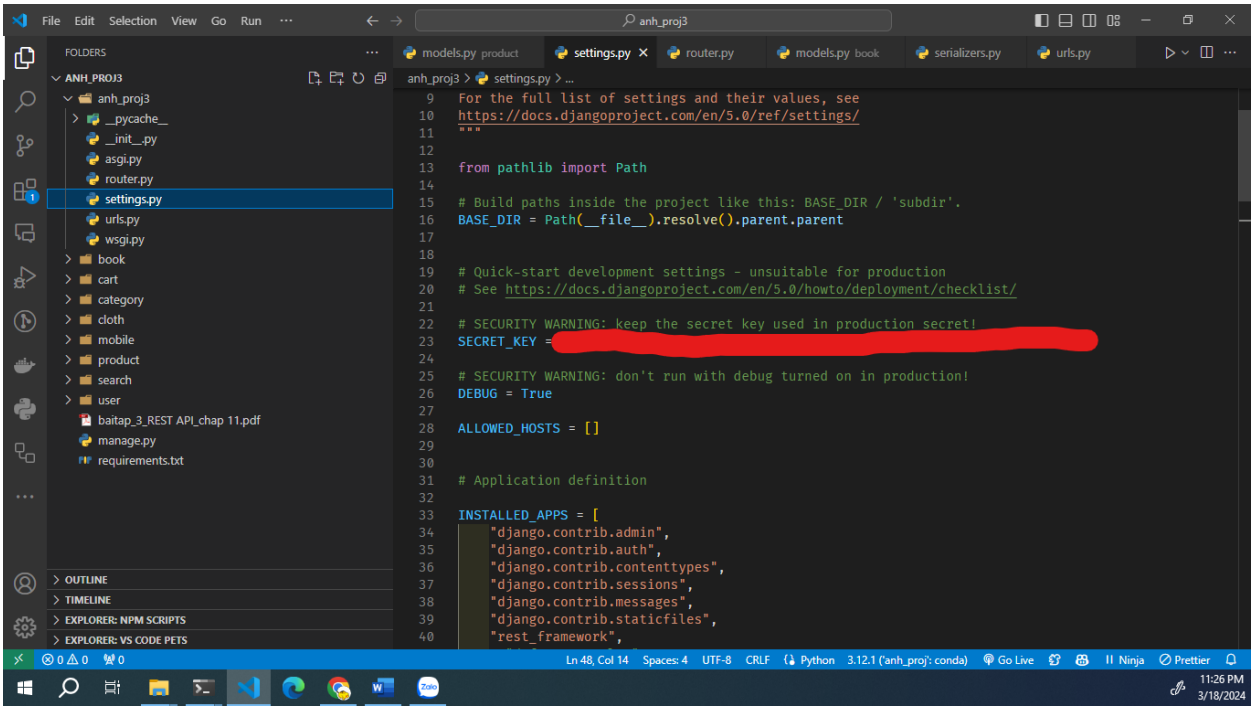


- Sau khi cài đặt các package trên, thực hiện tạo các service:
  - User: django-admin startapp user
  - Product: django-admin startapp product
  - Book: Django-admin startapp book
  - Cloth: Django-admin startapp cloth
  - Mobile: Django-admin startapp mobile
  - Category: Django-admin startapp category
  - Cart: Django-admin startapp cart
  - Search: Django-admin startapp search
  - Payment: Django-admin startapp payment
  - Shipment: Django-admin startapp shipment

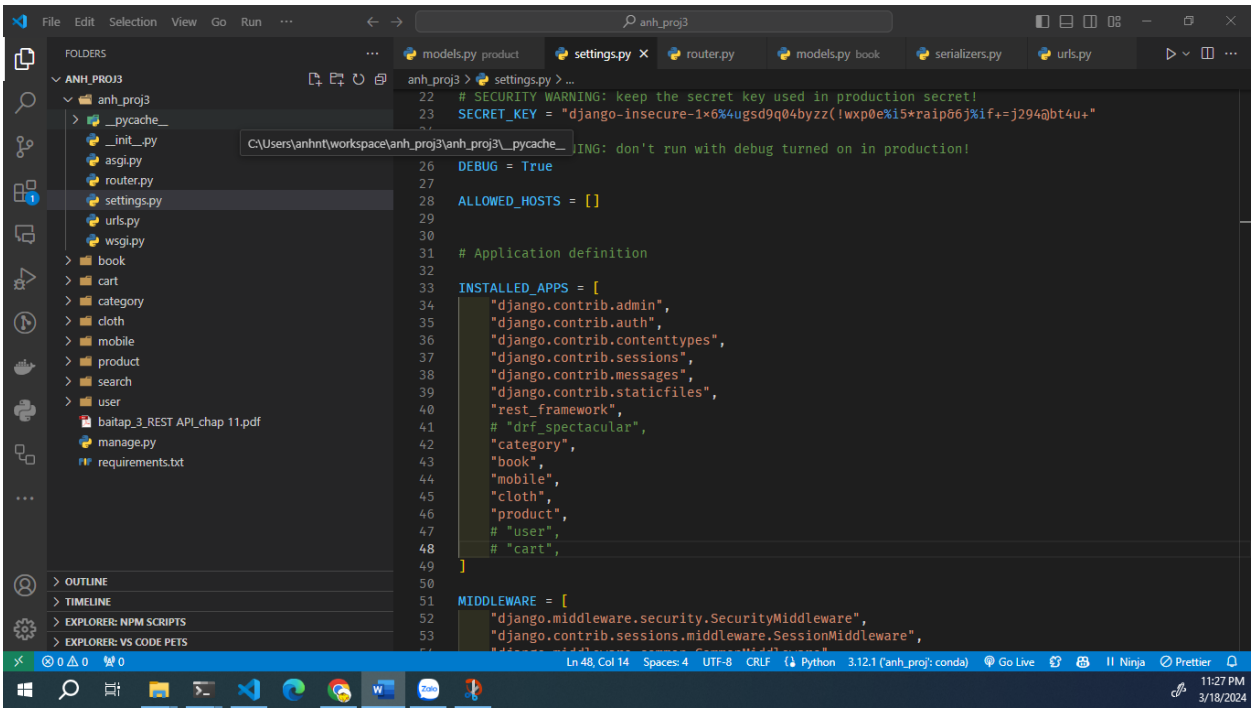


II. Cấu hình RESTful cho các API

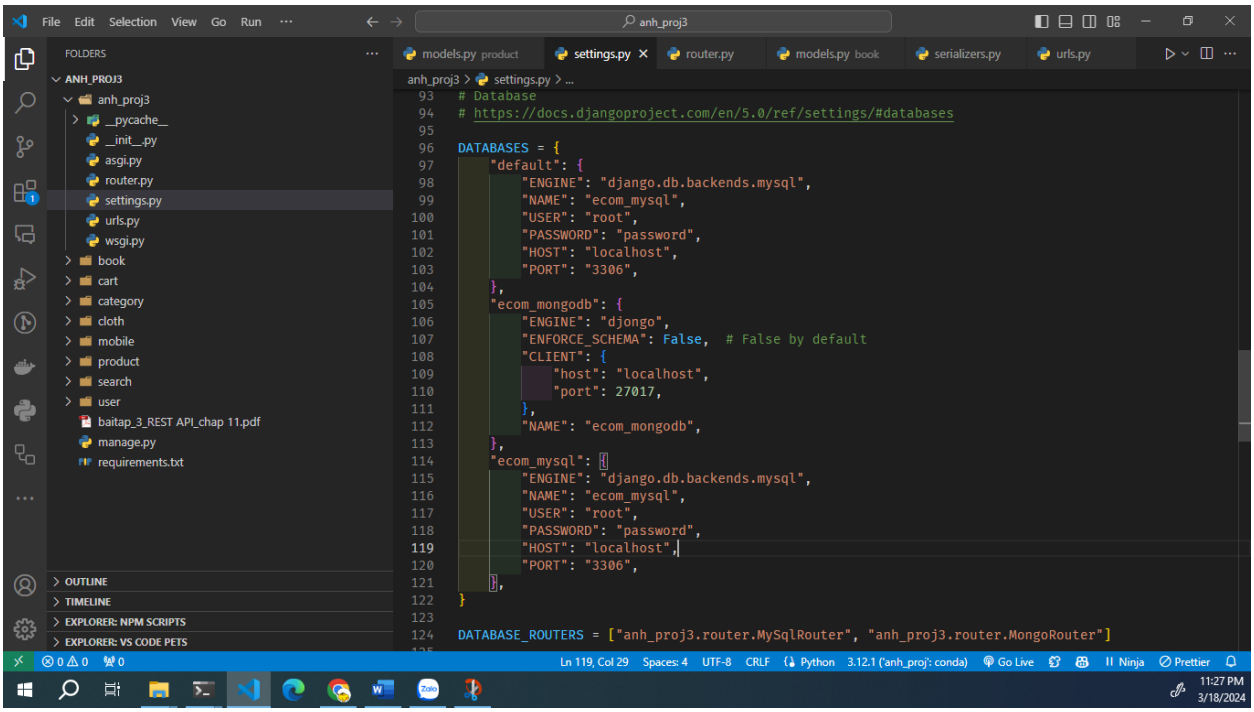
- Vào settings.py để tiện cho việc cấu hình:
  - Sửa SECRET_KEY cho việc mã hóa dữ liệu



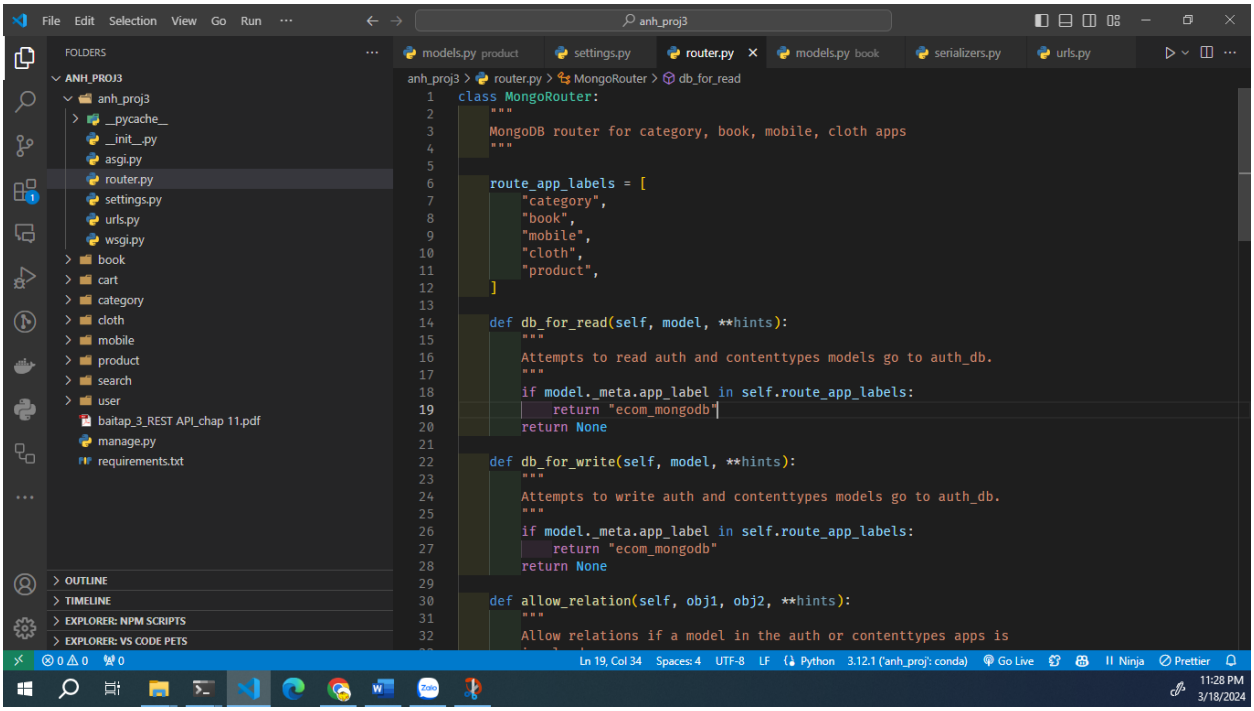
  - Thêm rest_framework vào INSTALLED_APPS

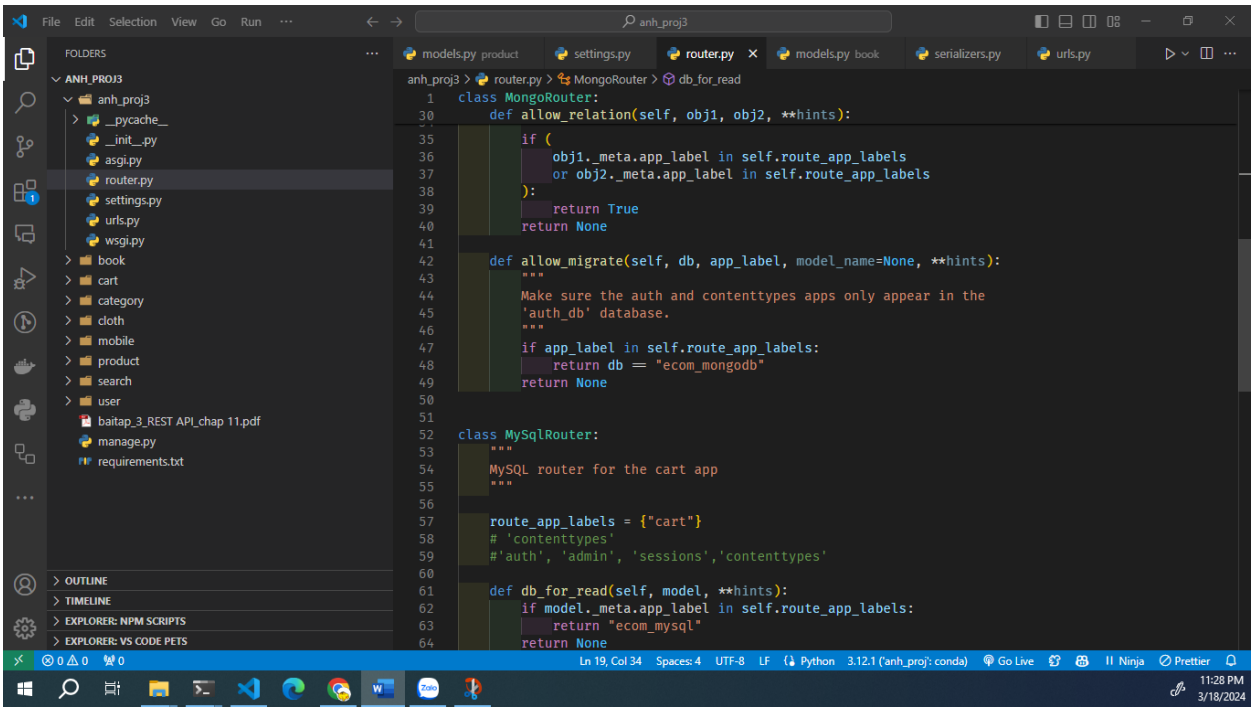


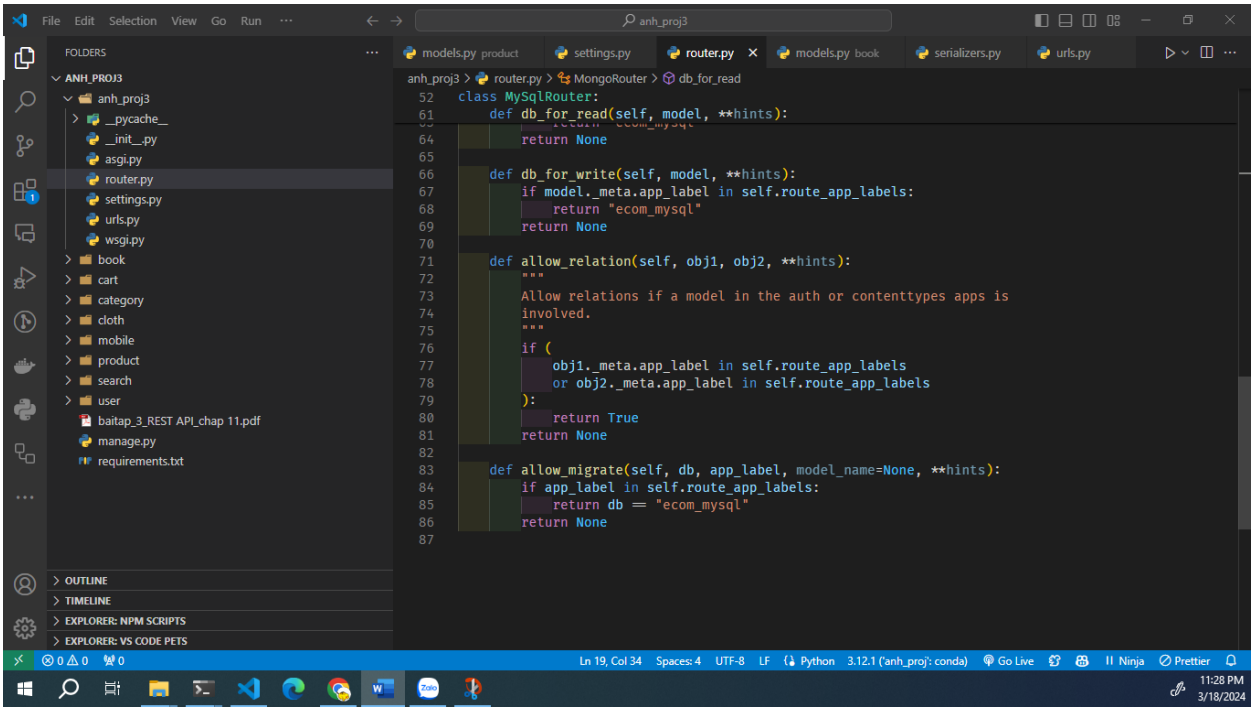
- Cấu hình database cho tiện cho việc phát triển app dựa trên MongoDB và MySQL.



- Tạo file routers.py để cấu hình điều hướng các app được tạo sao cho từng app dùng đúng loại database mà người dùng muốn.

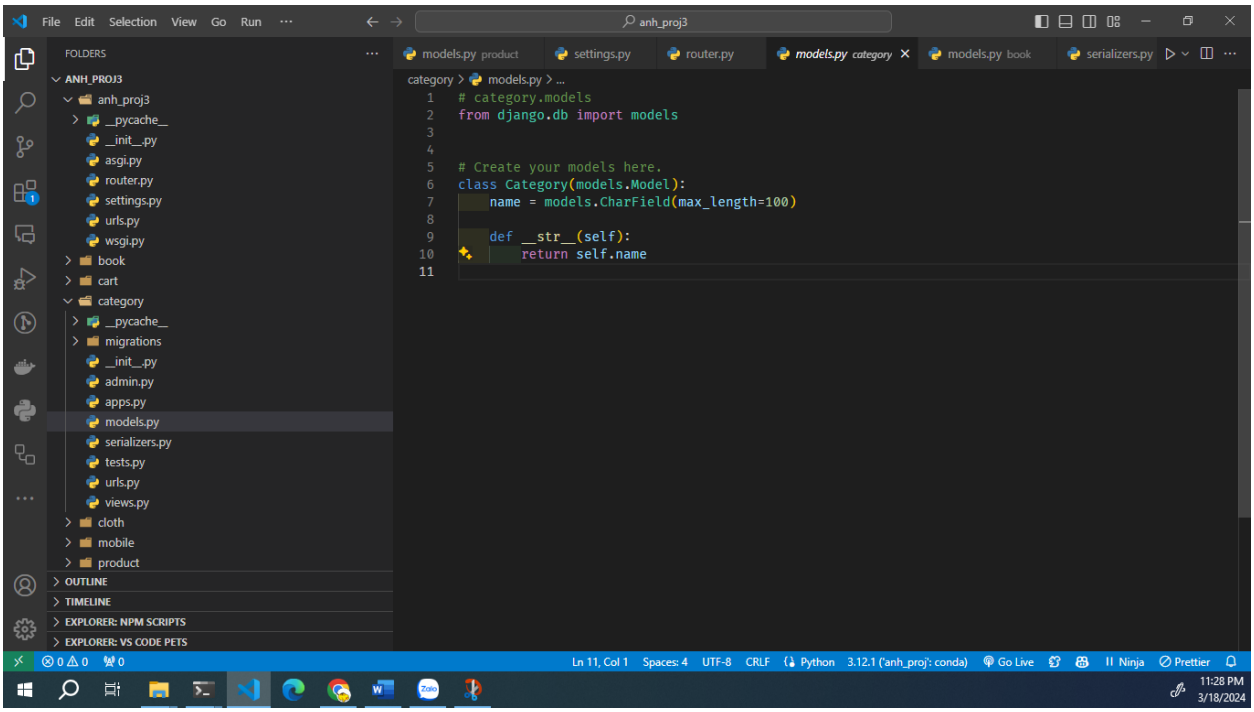

File Edit Selection View Go Run
anh_proj3
FOLDERS
ANH_PROJ3
anh_proj3
__pycache__
__init__.py
asgi.py
router.py
settings.py
urls.py
wsgi.py
book
cart
category
cloth
mobile
product
search
user
baitap_3_REST API_chap 11.pdf
manage.py
requirements.txt
models.py product
settings.py
router.py
models.py book
serializers.py
urls.py
anh_proj3 > router.py > MongoRouter > db_for_read
class MongoRouter:
"""
MongoDB router for category, book, mobile, cloth apps
"""
route_app_labels = [
"category",
"book",
"mobile",
"cloth",
"product",
]
def db_for_read(self, model, **hints):
"""
Attempts to read auth and contenttypes models go to auth_db.
"""
if model._meta.app_label in self.route_app_labels:
return "ecom_mongodb"
return None
def db_for_write(self, model, **hints):
"""
Attempts to write auth and contenttypes models go to auth_db.
"""
if model._meta.app_label in self.route_app_labels:
return "ecom_mongodb"
return None
def allow_relation(self, obj1, obj2, **hints):
"""
Allow relations if a model in the auth or contenttypes apps is
OUTLINE
TIMELINE
EXPLORER: NPM SCRIPTS
EXPLORER: VS CODE PETS
Ln 19, Col 34 Spaces: 4 UTF-8 LF Python 3.12.1 ('anh_proj': conda) Go Live Ninja Prettier
11:28 PM
3/18/2024


File Edit Selection View Go Run
anh_proj3
FOLDERS
ANH_PROJ3
anh_proj3
__pycache__
__init__.py
asgi.py
router.py
settings.py
urls.py
wsgi.py
book
cart
category
cloth
mobile
product
search
user
baitap_3_REST API_chap 11.pdf
manage.py
requirements.txt
models.py product
settings.py
router.py
models.py book
serializers.py
urls.py
anh_proj3 > router.py > MongoRouter > db_for_read
class MongoRouter:
def allow_relation(self, obj1, obj2, **hints):
if (
obj1._meta.app_label in self.route_app_labels
or obj2._meta.app_label in self.route_app_labels
):
return True
return None
def allow_migrate(self, db, app_label, model_name=None, **hints):
"""
Make sure the auth and contenttypes apps only appear in the
'auth_db' database.
"""
if app_label in self.route_app_labels:
return db == "ecom_mongodb"
return None
class MySqlRouter:
"""
MySQL router for the cart app
"""
route_app_labels = {"cart"}
# 'contenttypes'
#'auth', 'admin', 'sessions','contenttypes'
def db_for_read(self, model, **hints):
if model._meta.app_label in self.route_app_labels:
return "ecom_mysql"
return None
OUTLINE
TIMELINE
EXPLORER: NPM SCRIPTS
EXPLORER: VS CODE PETS
Ln 19, Col 34 Spaces: 4 UTF-8 LF Python 3.12.1 ('anh_proj': conda) Go Live Ninja Prettier
11:28 PM
3/18/2024

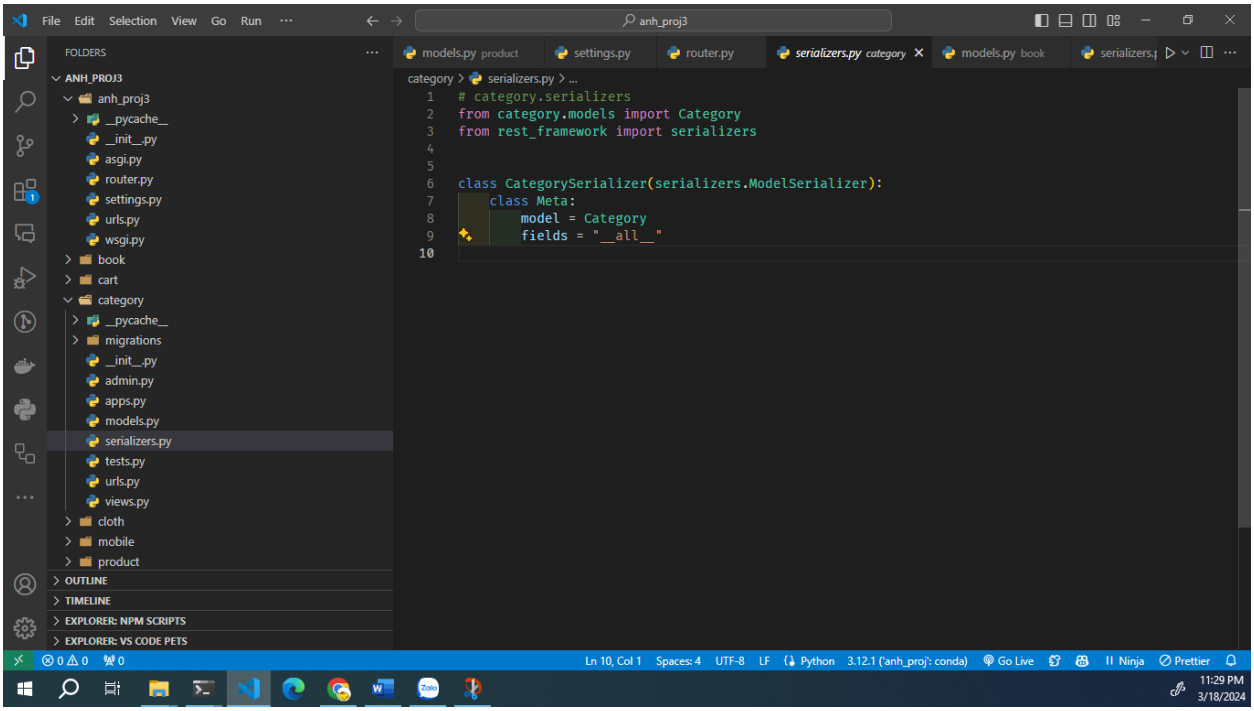


- Thêm chỉnh sửa CORS thành "*" để tiện cho việc giao diện gọi API

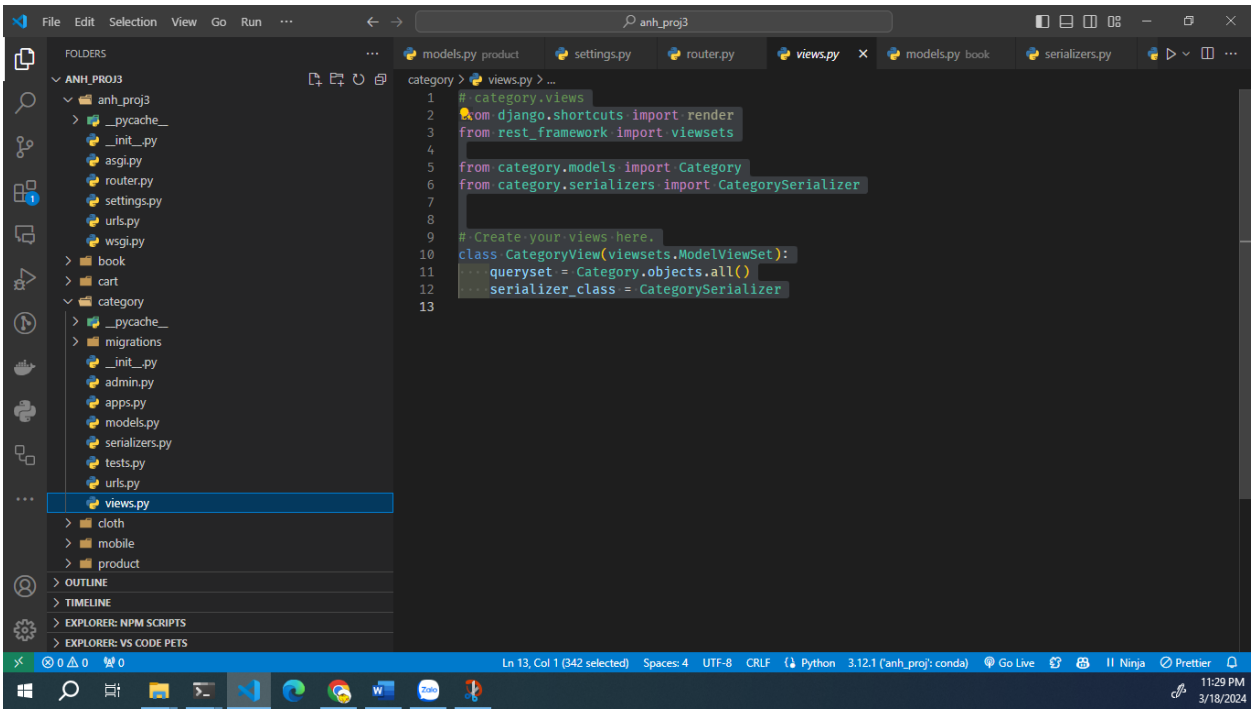
- Category:
  - Tạo model cho Category trong models.py



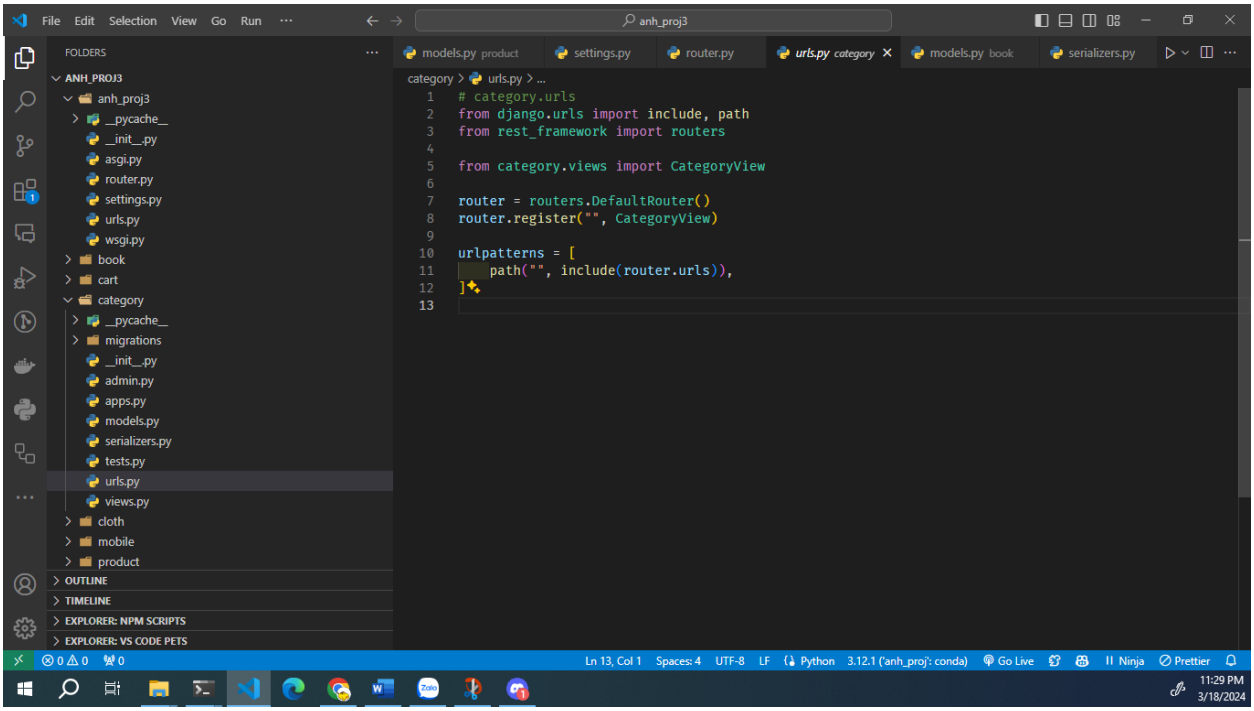
  - Tạo serializers cho Category trong serializers.py



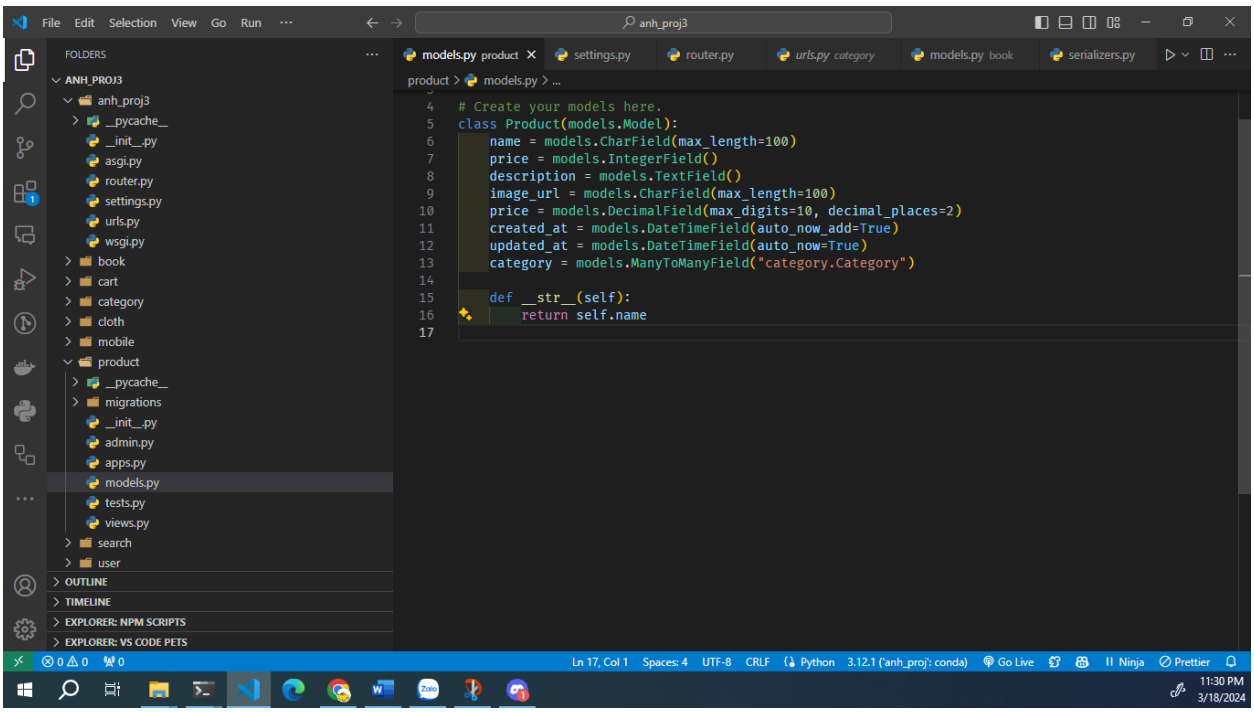
- Tạo views cho category trong views.py



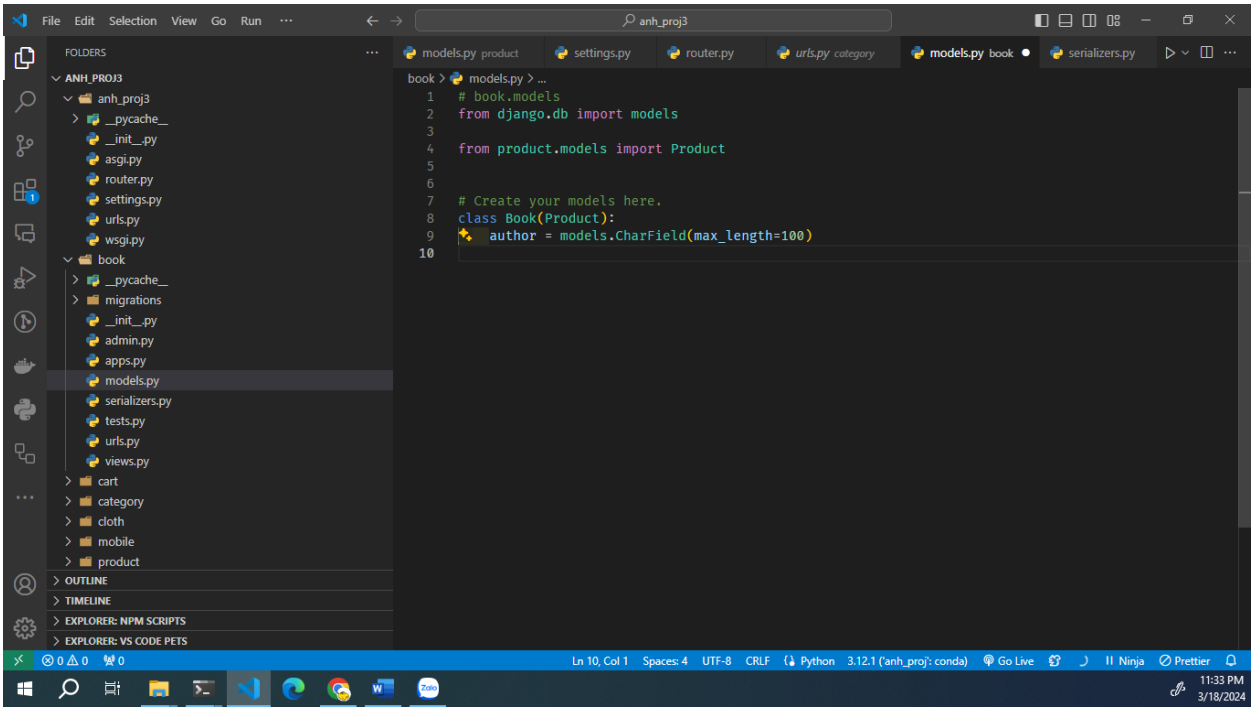
- Tạo urlpatterns cho category trong urls.py



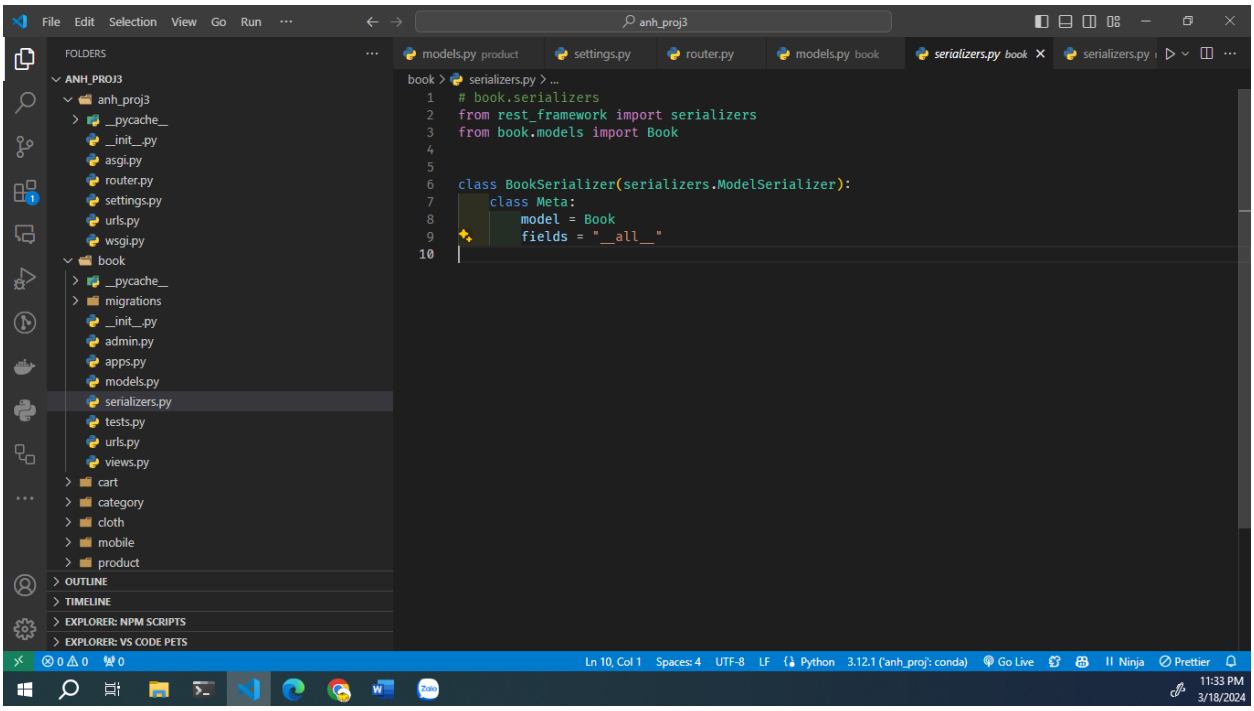
- Product:
  - Tạo model cho Product trong models.py



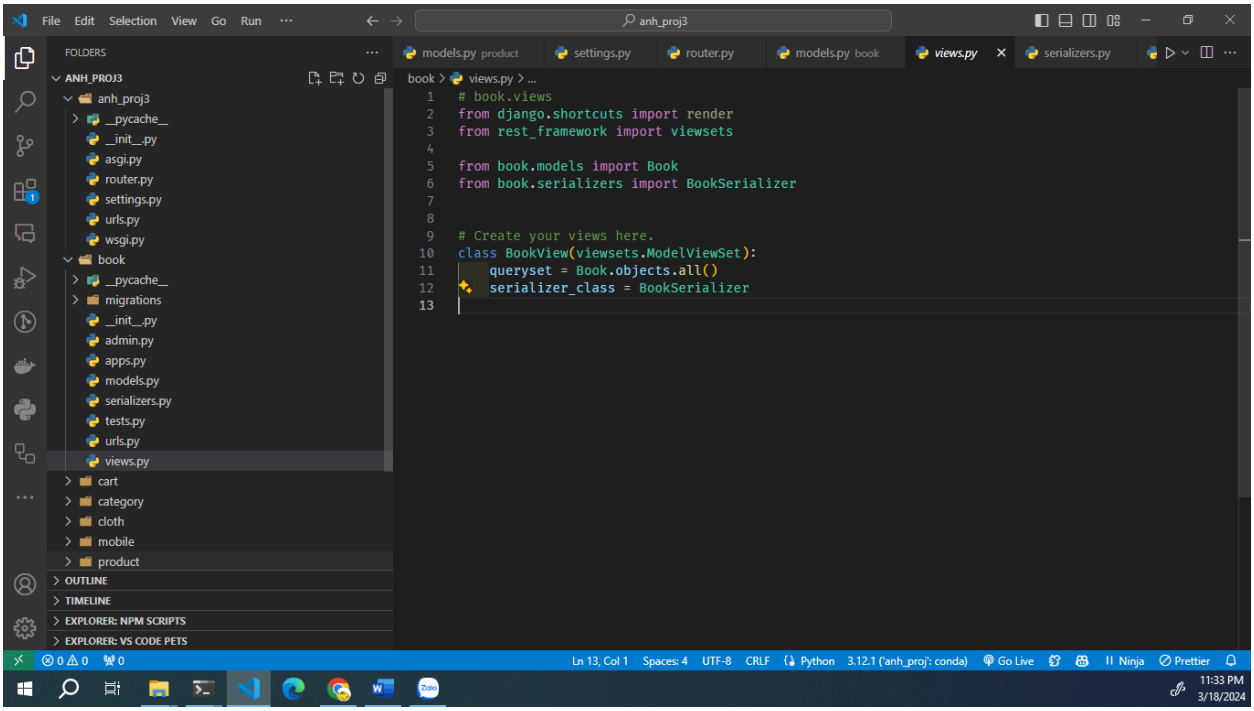
- Book:
  - Tạo model cho book extends từ product trong models.py



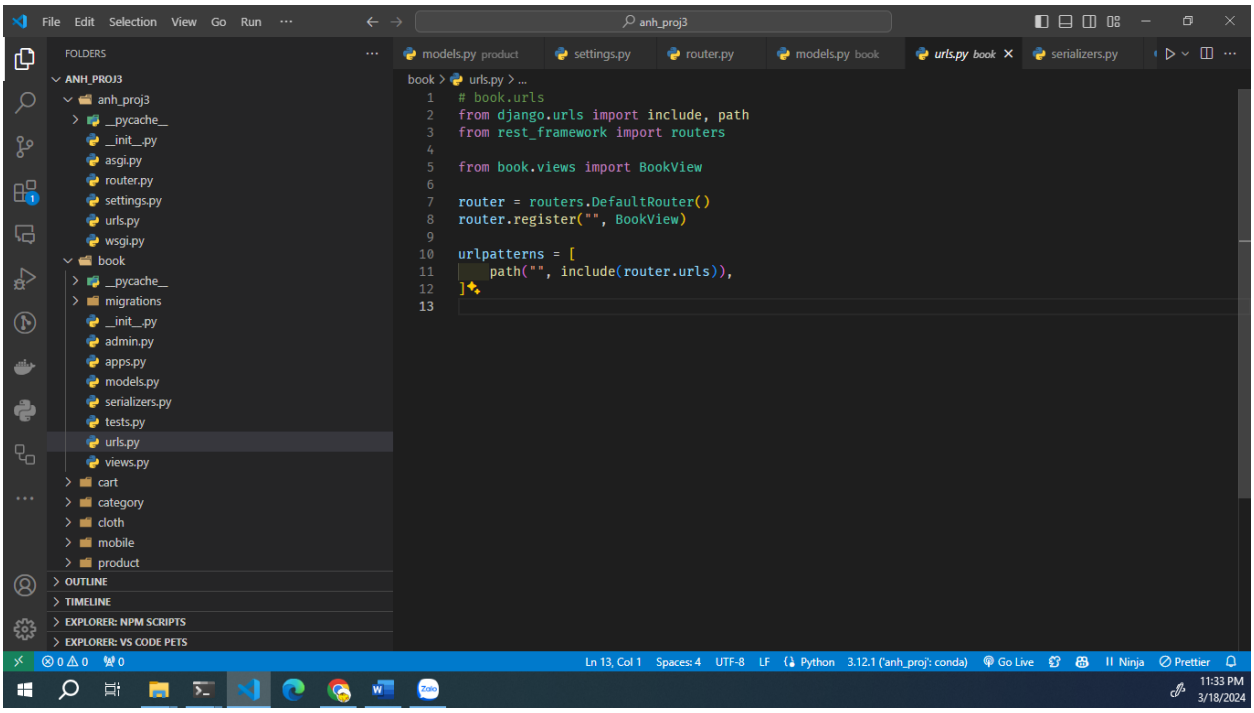
- Tạo serializers cho book trong serializers.py



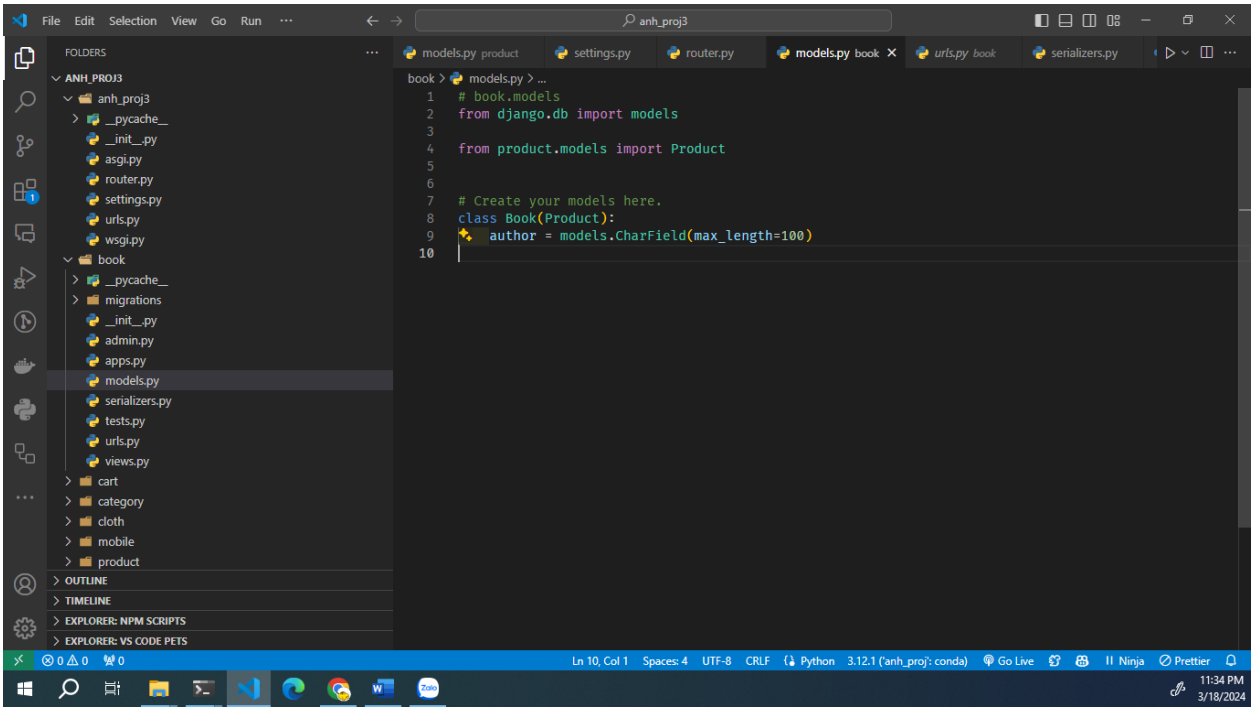
- Tạo views cho book trong views.py



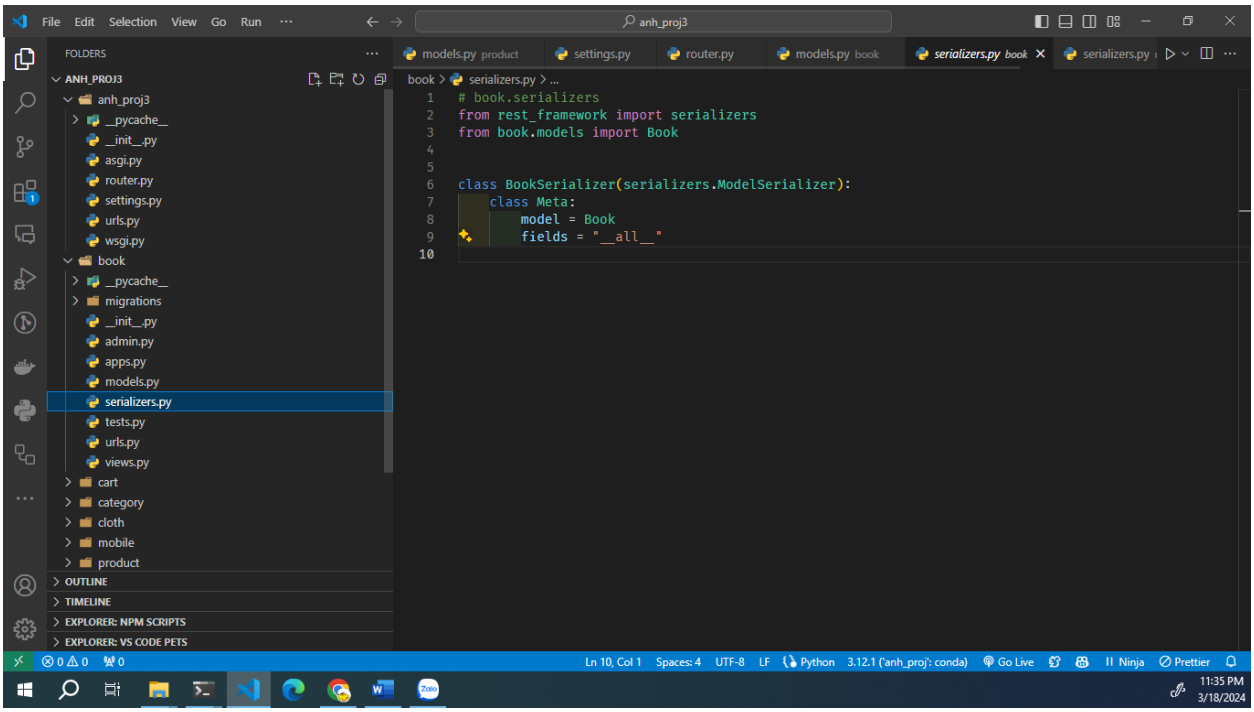
- Tạo urlpatterns cho book trong urls.py



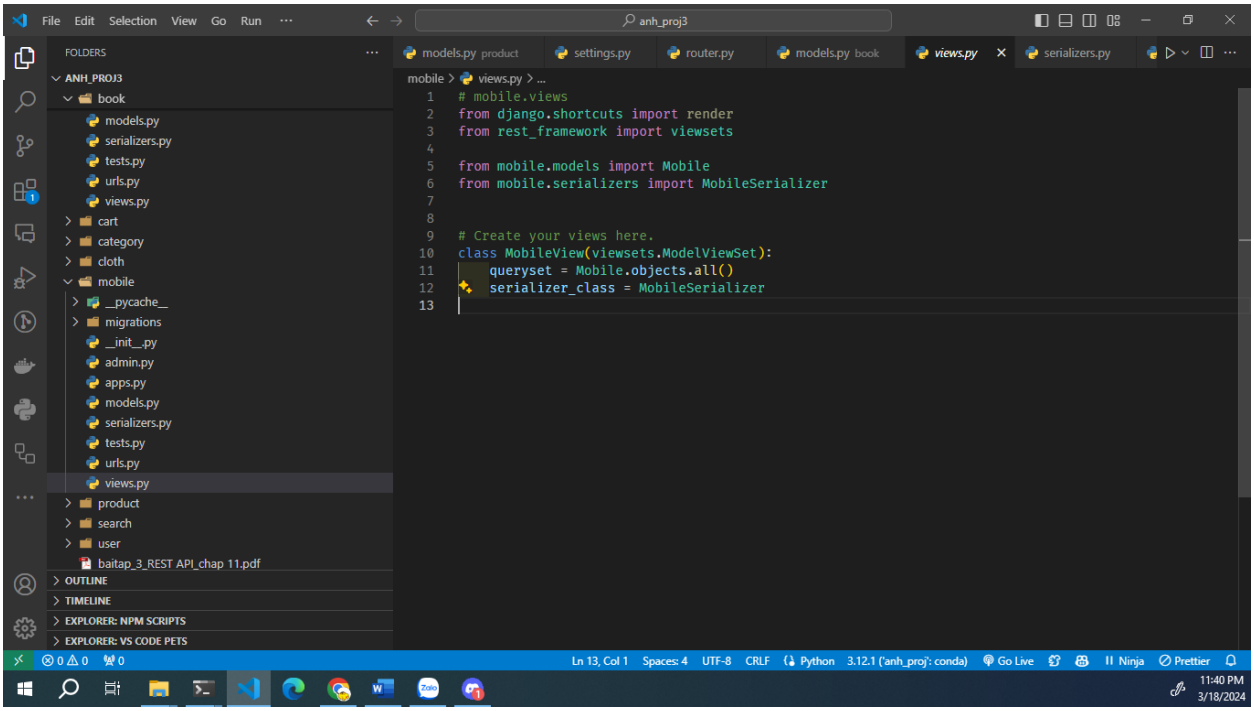
- Mobile:
  - Tạo model cho mobile extends từ product trong models.py



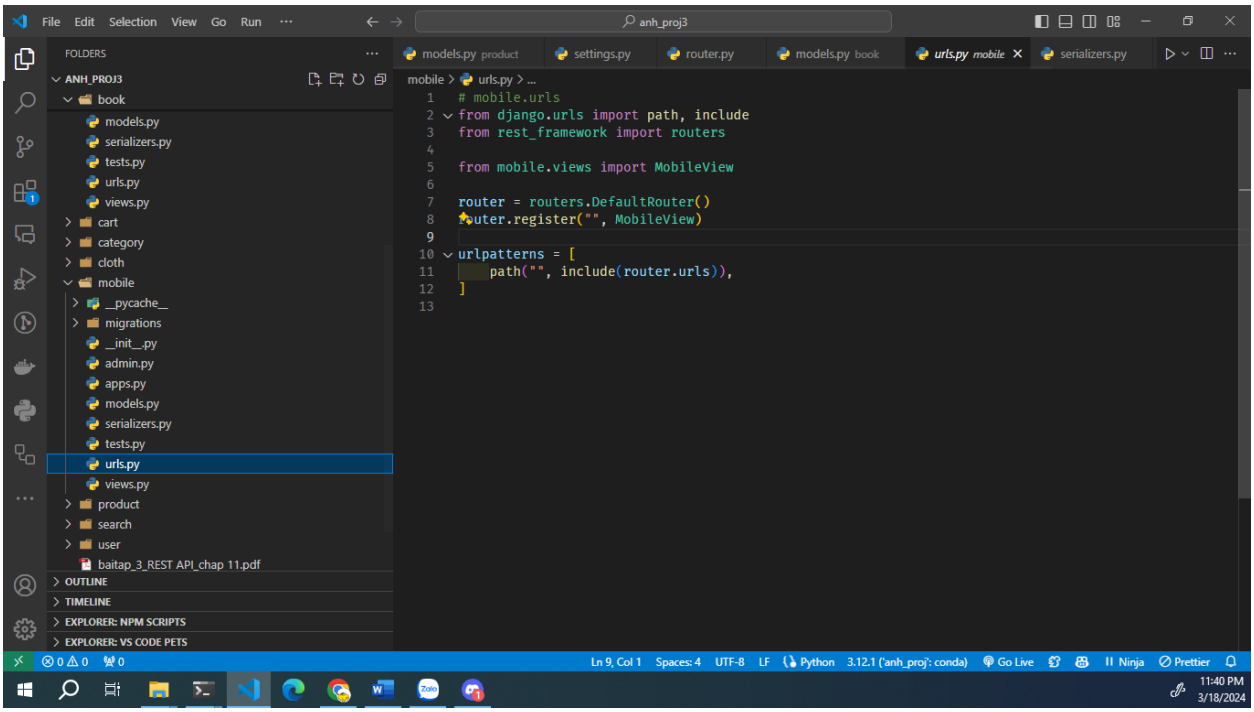
- Tạo serializers cho mobile trong serializers.py



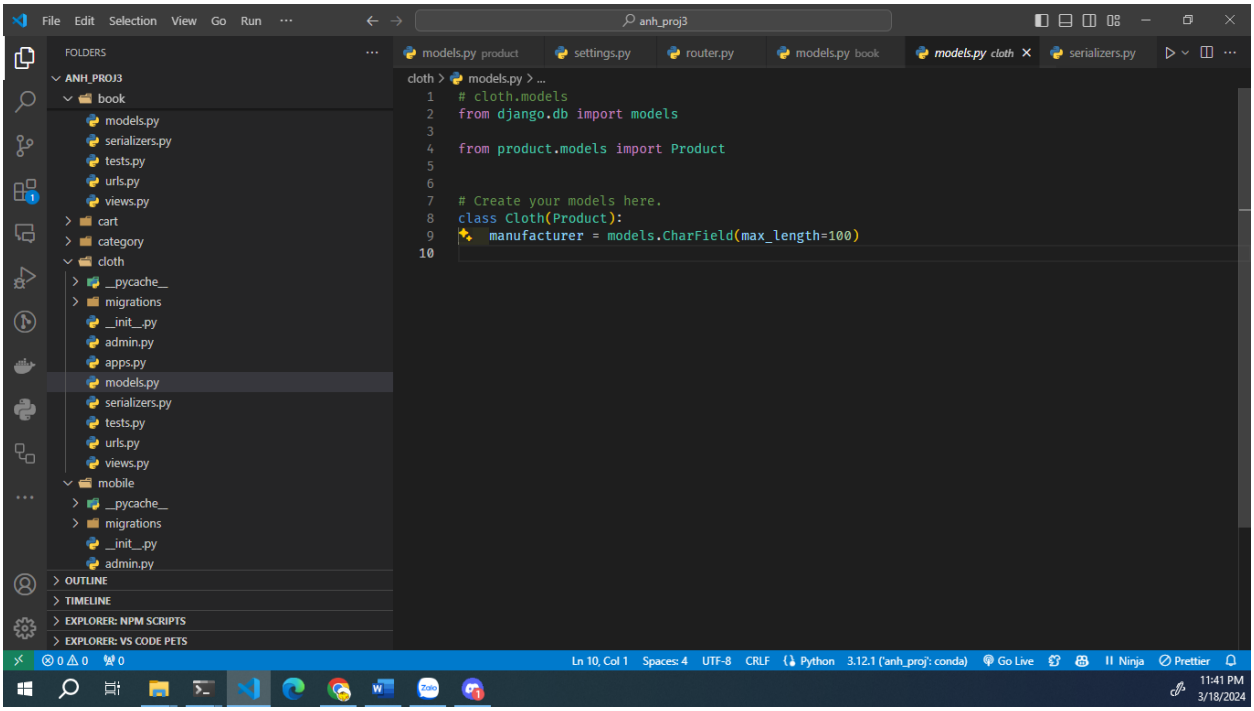
- Tạo views cho mobile trong views.py



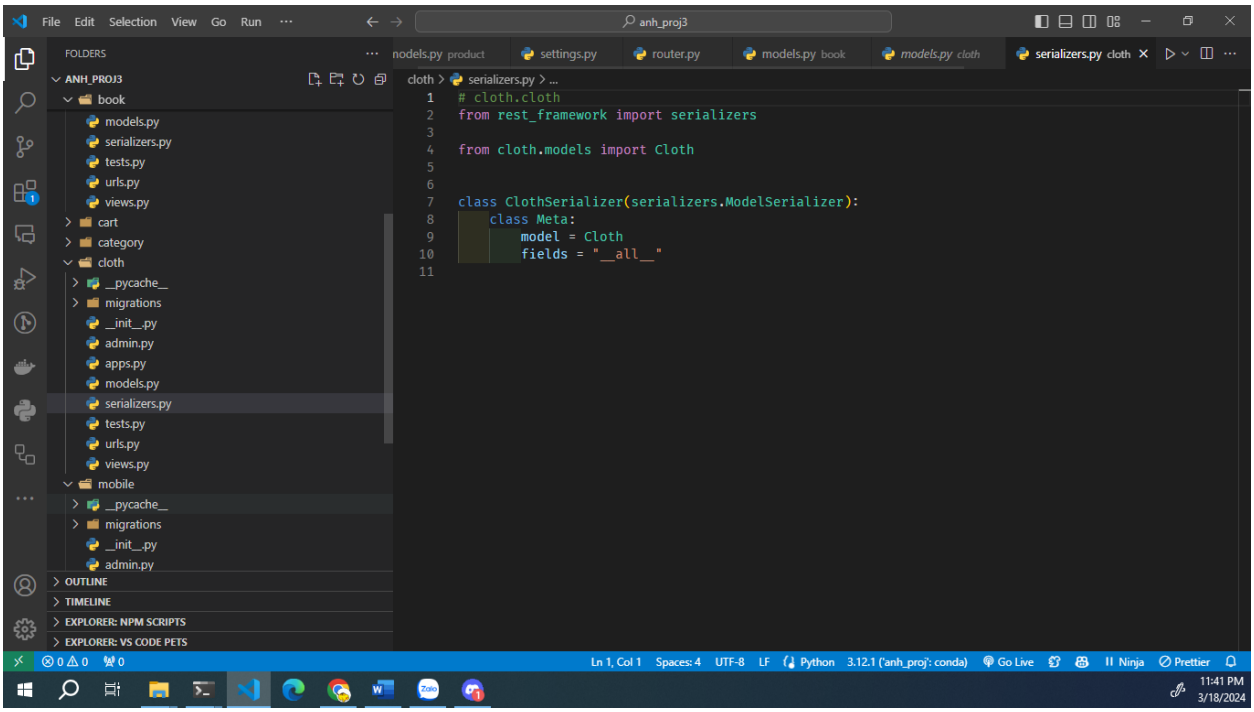
- Tạo urlpatterns cho mobile trong urls.py



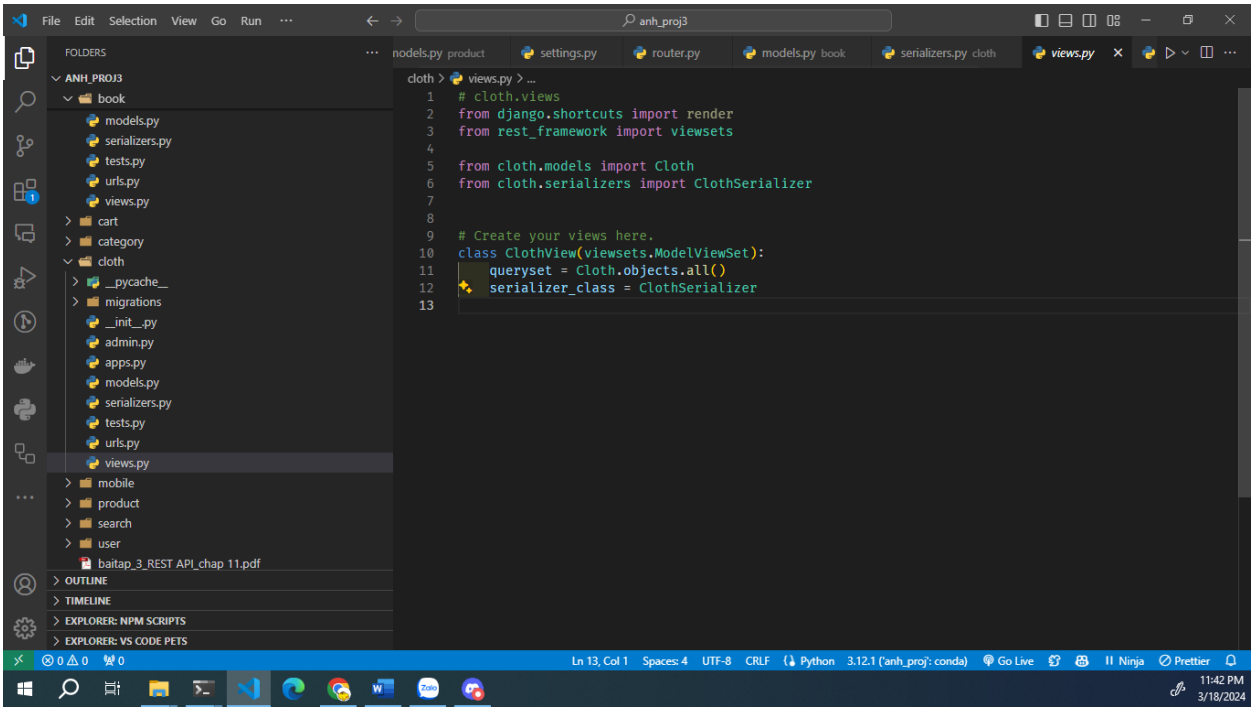
- Cloth:
  - Tạo model cho cloth trong models.py



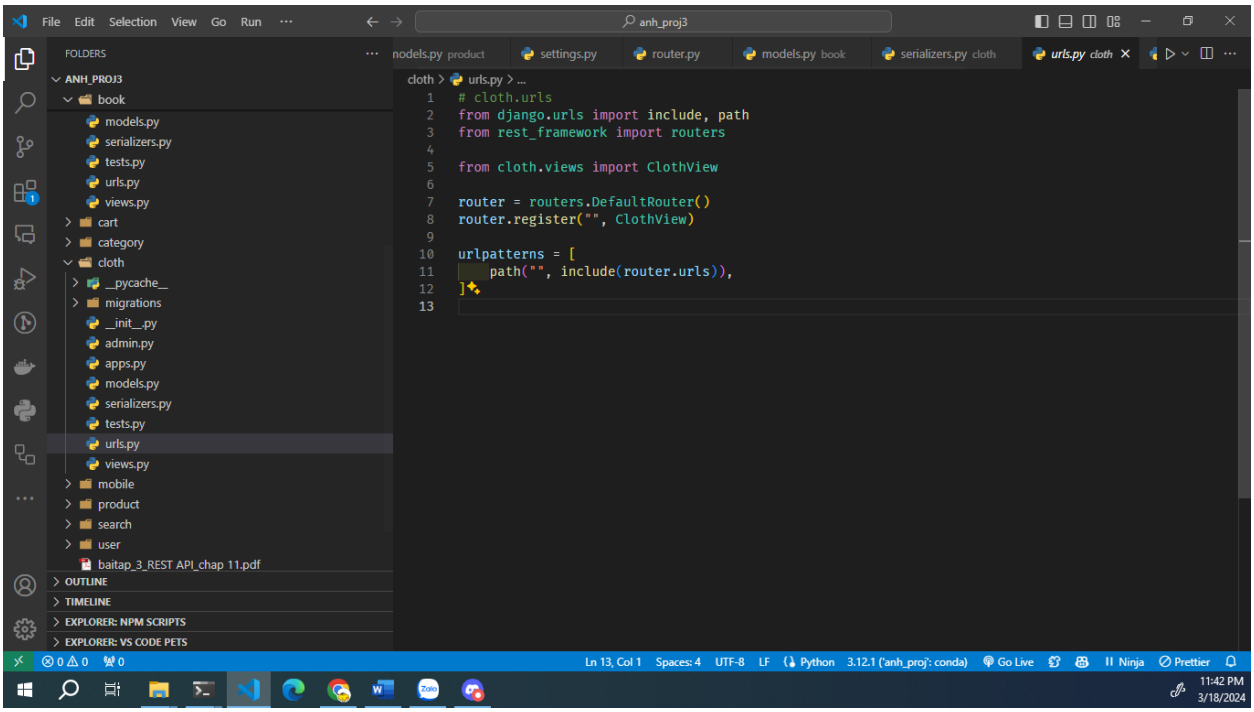
- Tạo serializers cho cloth trong serializers.py



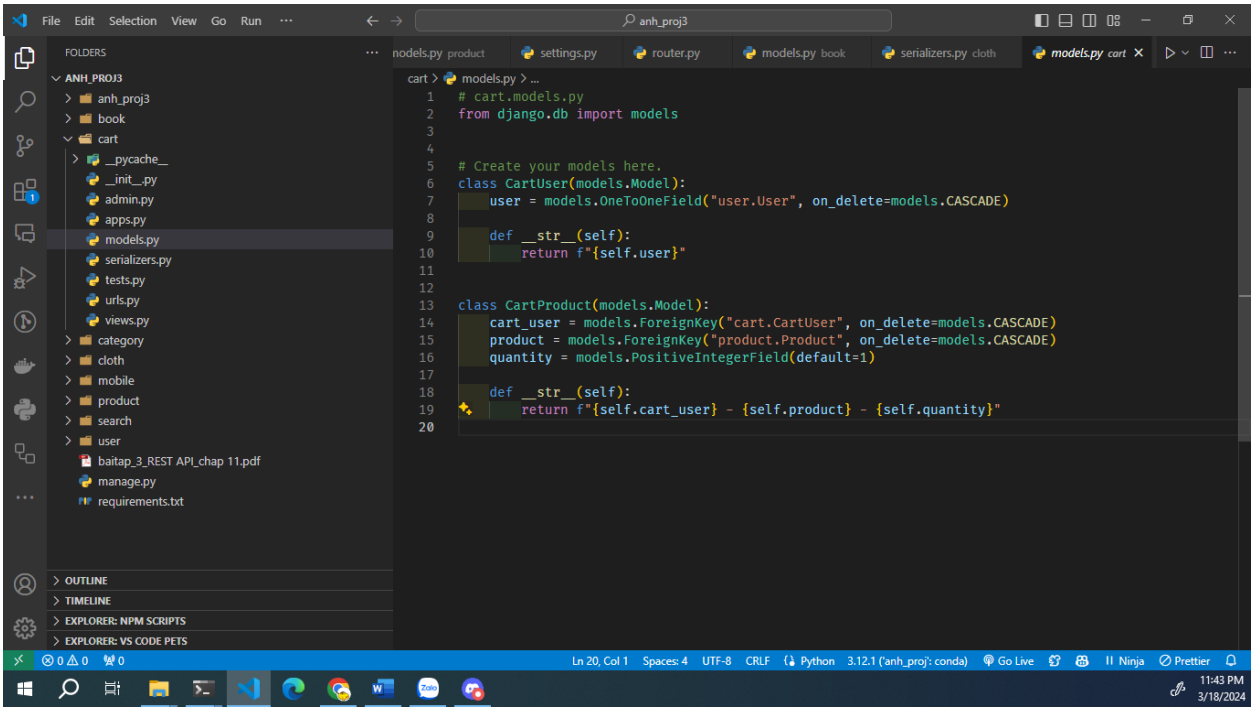
- Tạo views cho cloth trong views.py



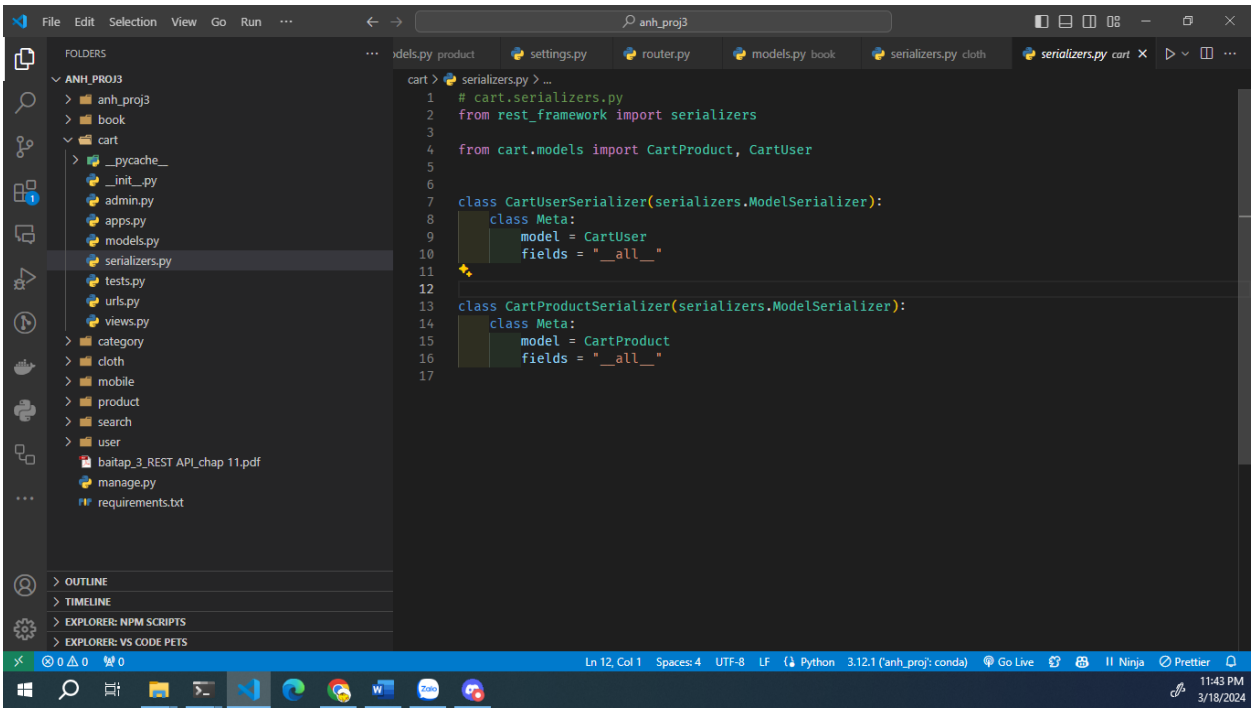
- Tạo urlpatterns cho cloth trong urls.py



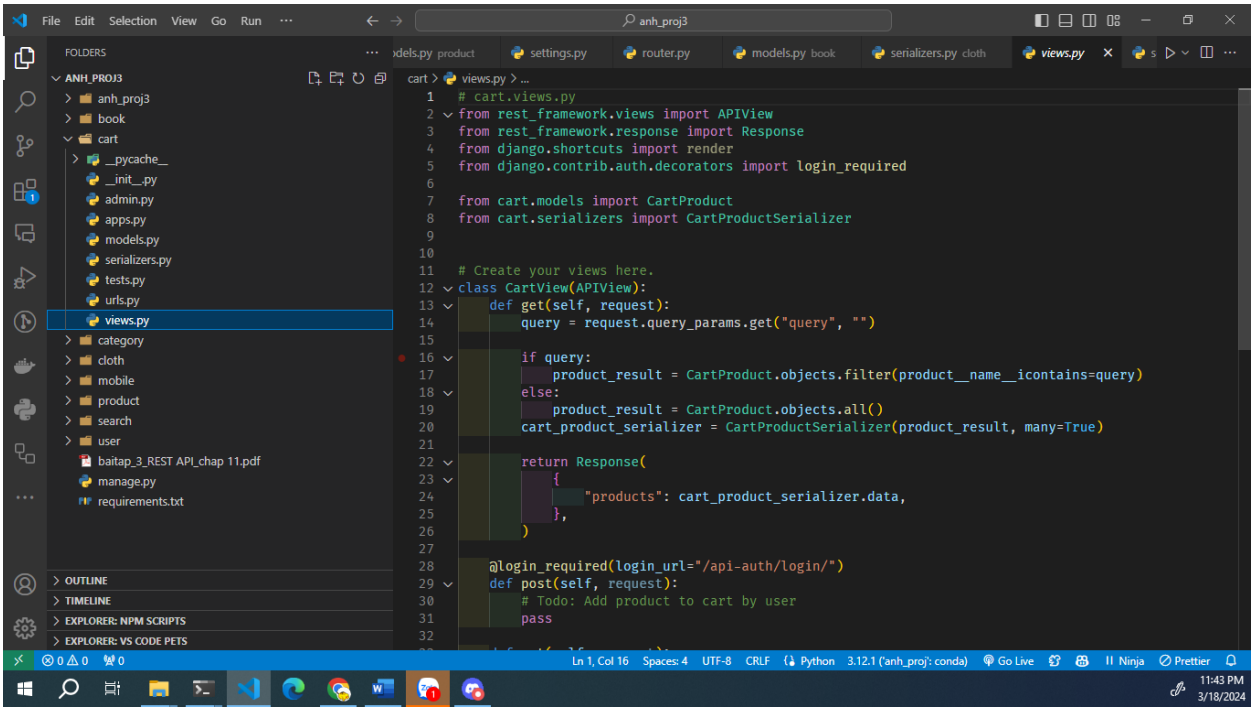
- Cart:
  - Tạo model cho cart trong models.py



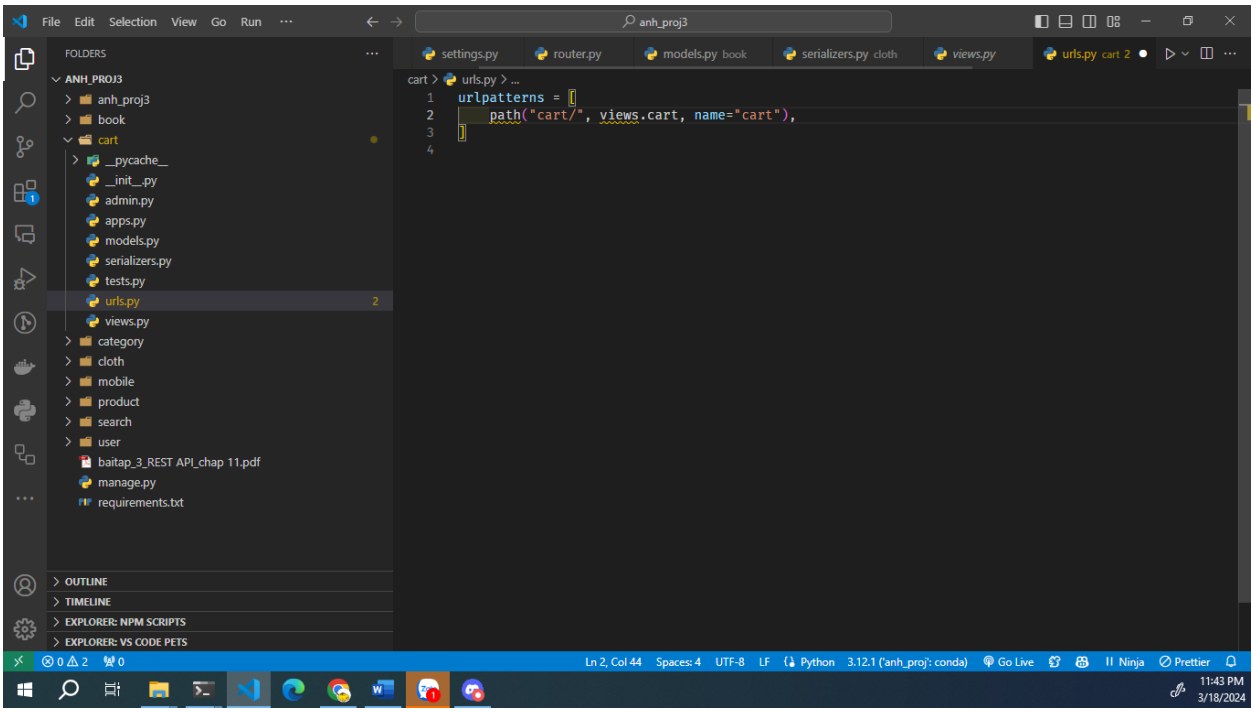
- Tạo serializers cho cart trong serializers.py



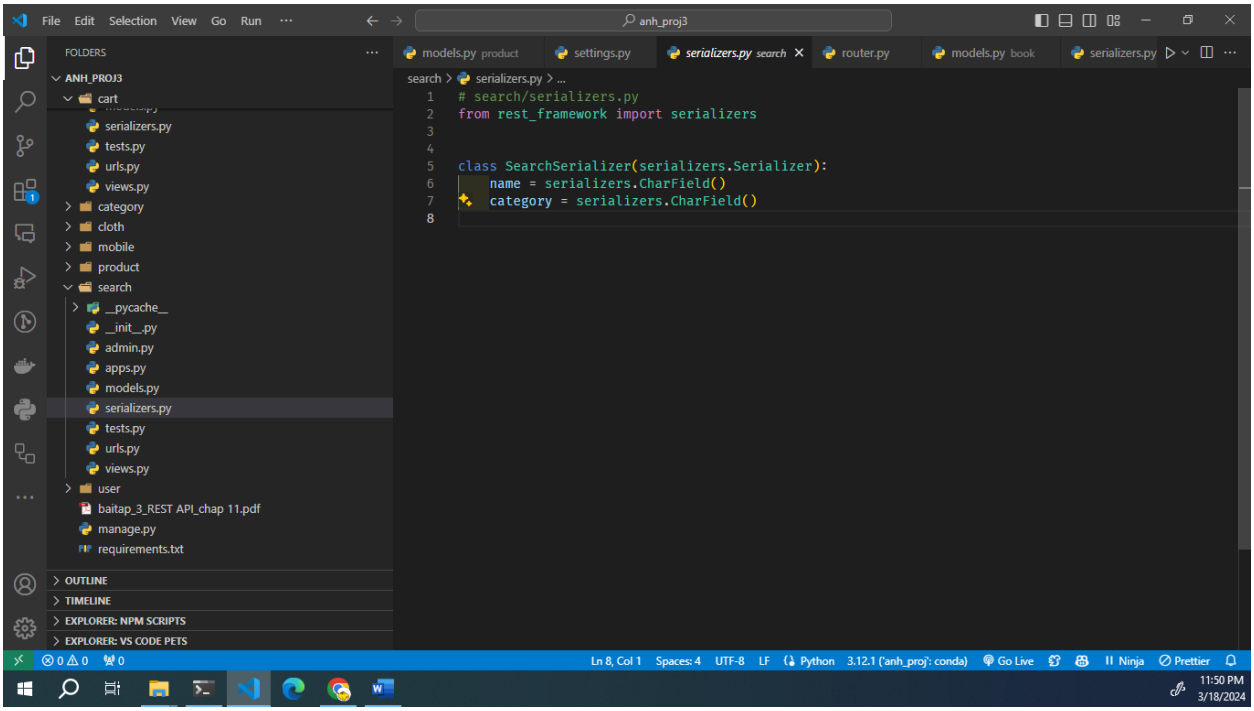
- Tạo views cho cart trong views.py



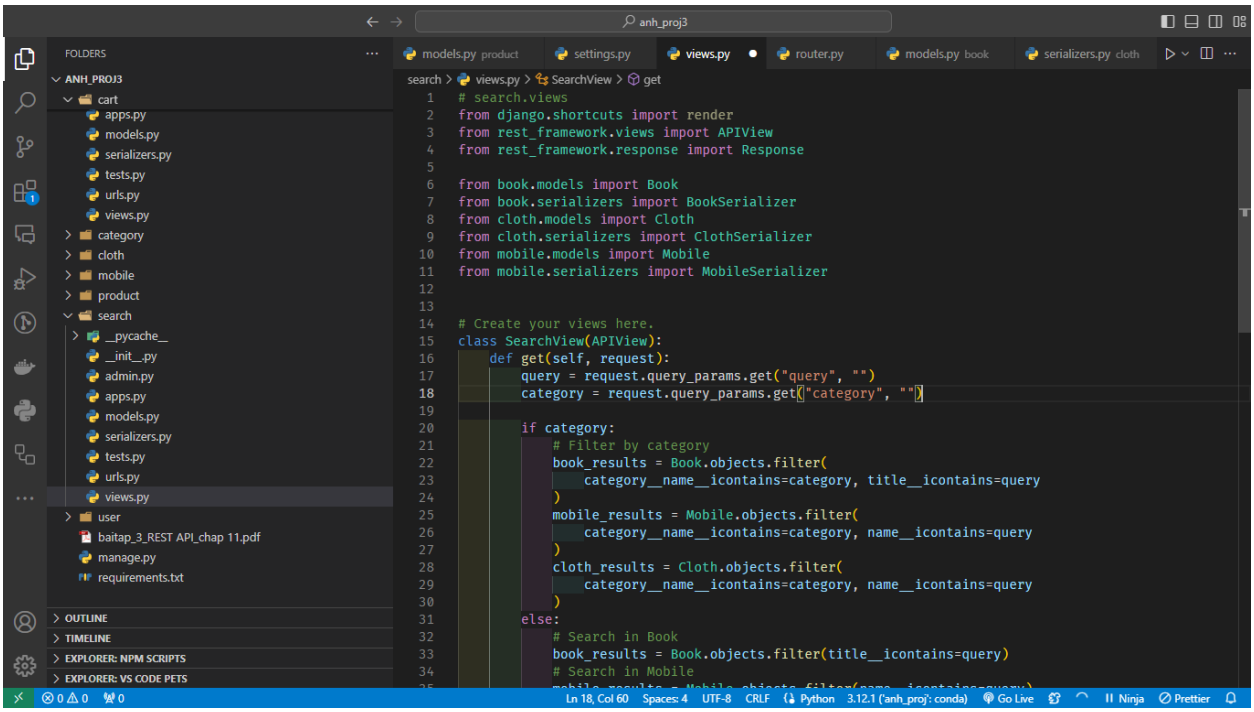
- Tạo urlpatterns cho cart trong urls.py

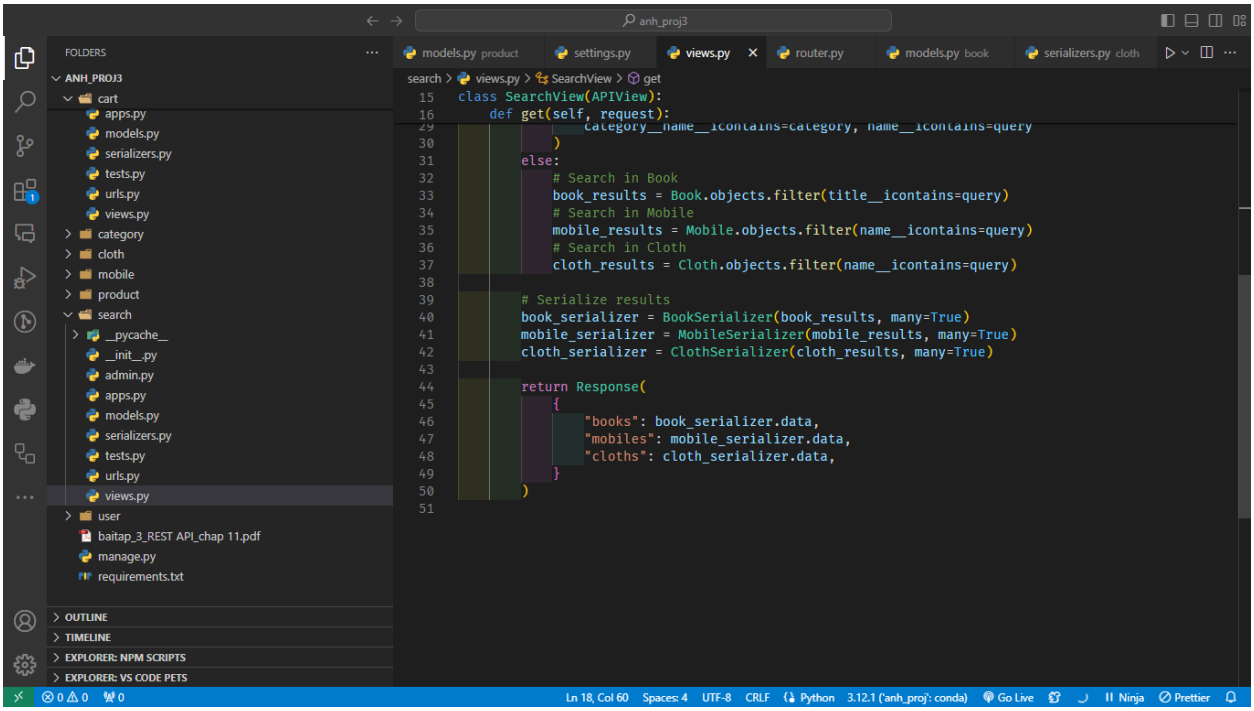


- Search:
  - Tạo serializers cho search

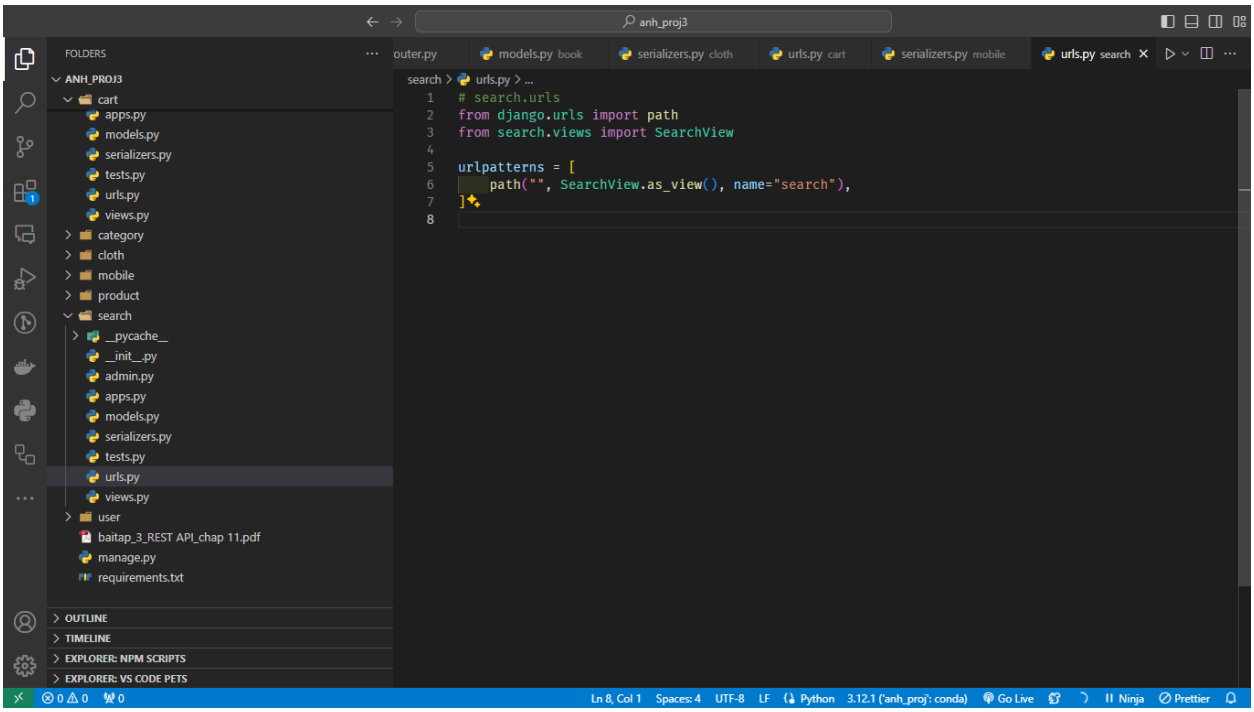


- Tạo views cho search

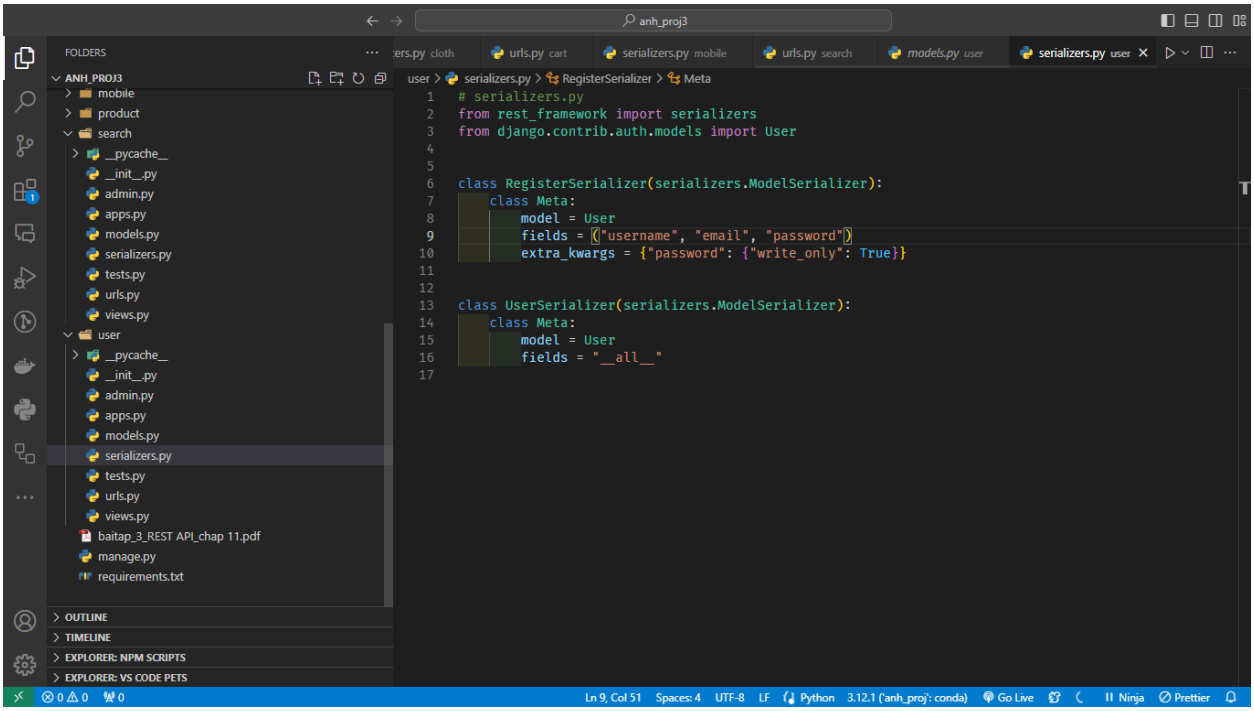




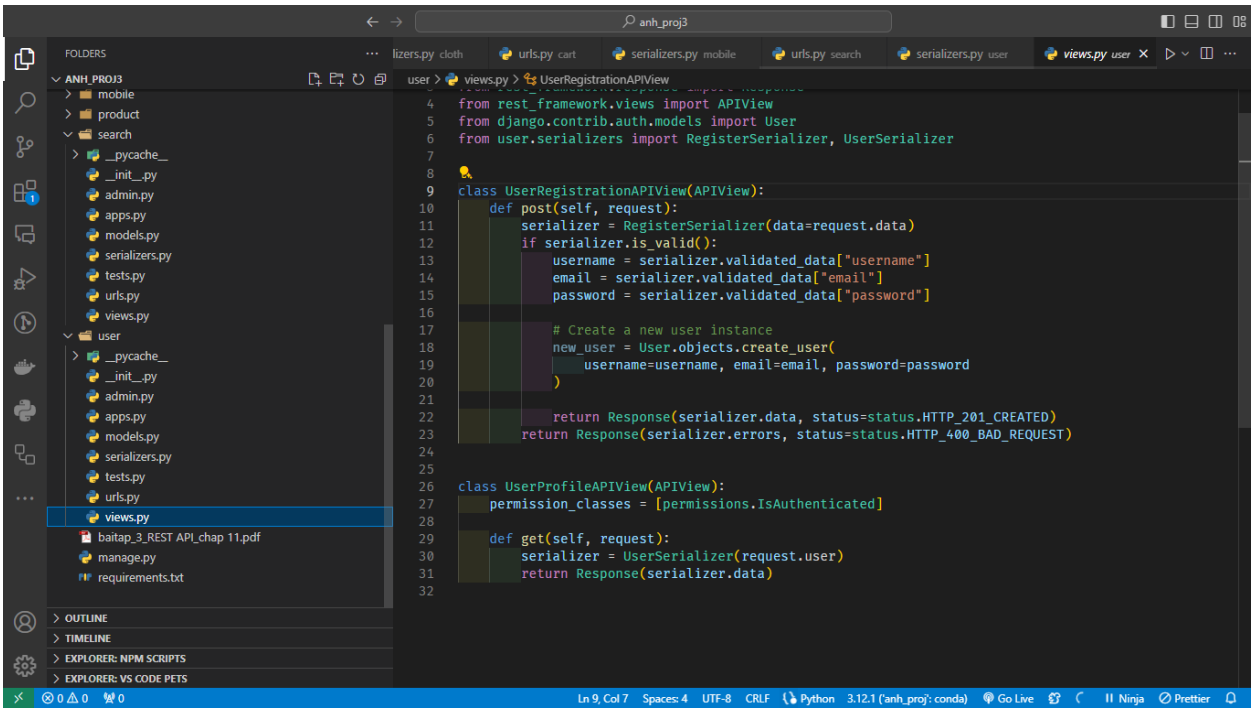
- Tạo urlpatterns cho search



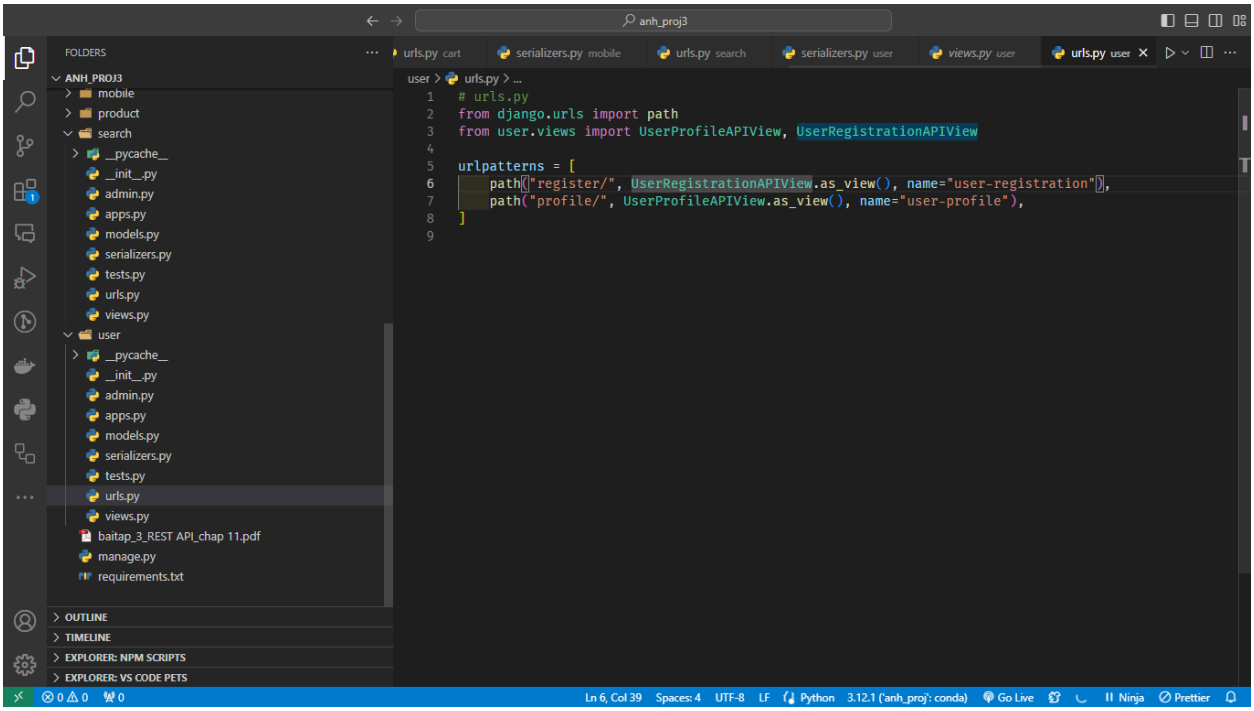
- User:
  - Tạo serializers cho user



- Tạo views cho user

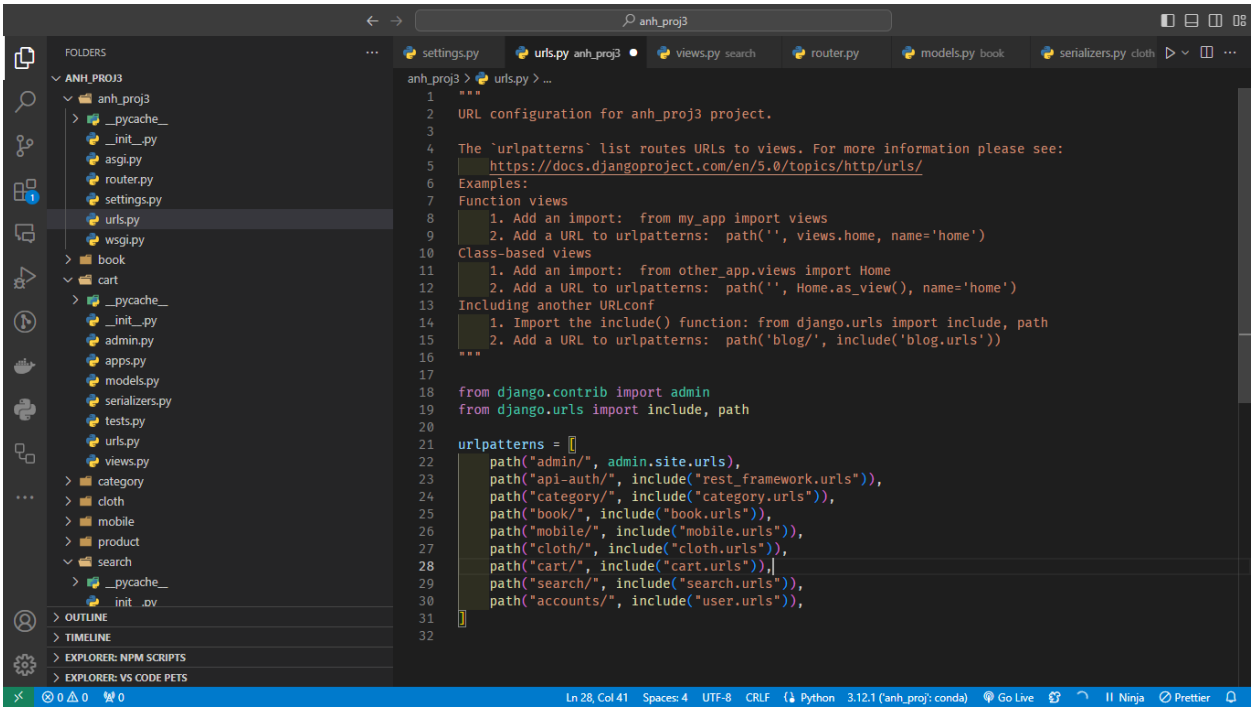


- Tạo urlpatterns cho user

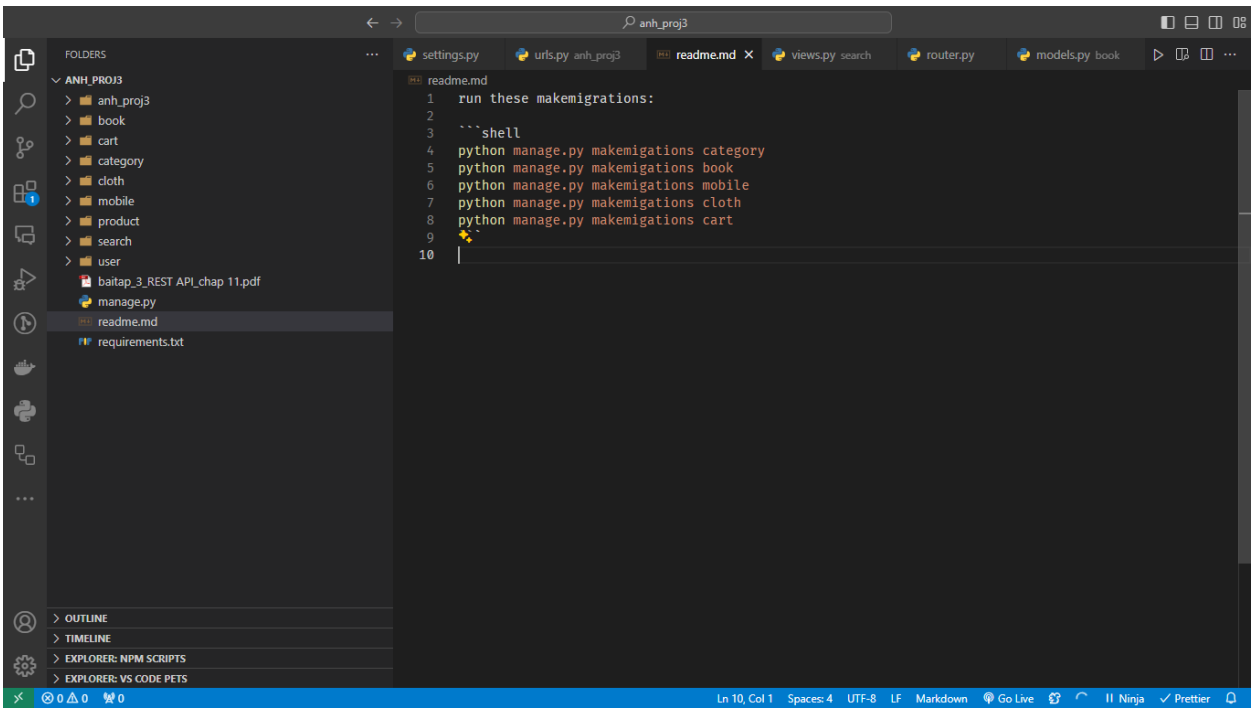


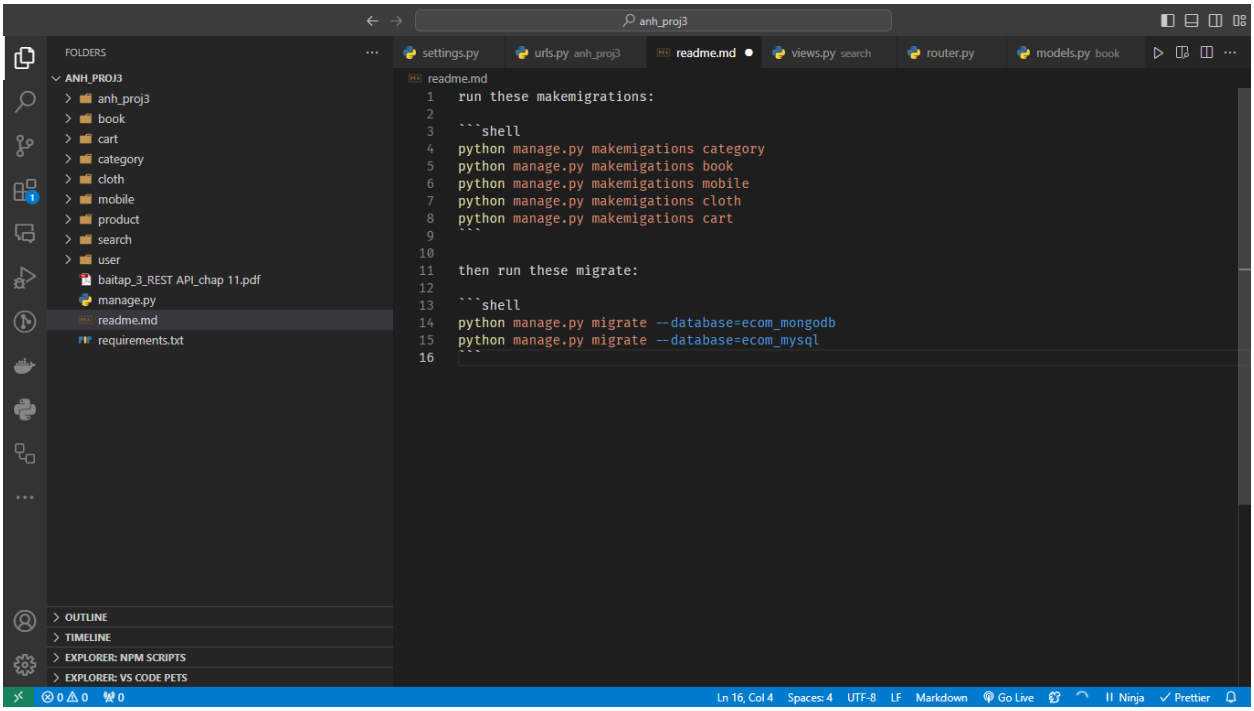
- Register các app trên vào INSTALLED_APPS và urls.py ở root

```
INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "rest_framework",
    # "drf_spectacular",
    "category",
    "book",
    "mobile",
    "cloth",
    "product",
    "user",
    "cart",
]
```

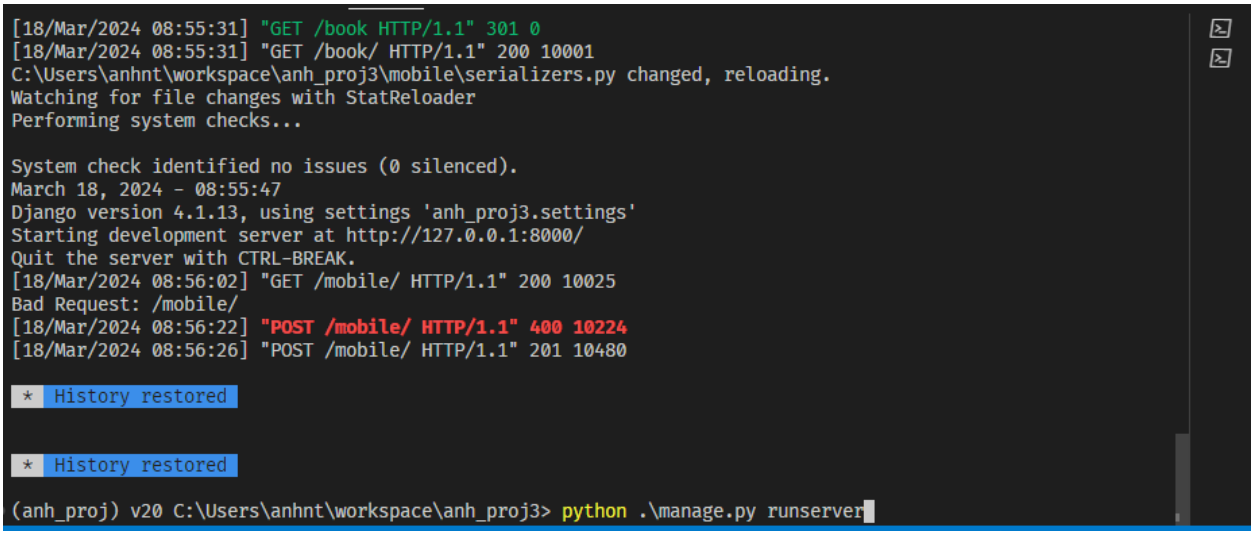


- Chạy câu lệnh migration cho các database và makemigrations cho các app model.

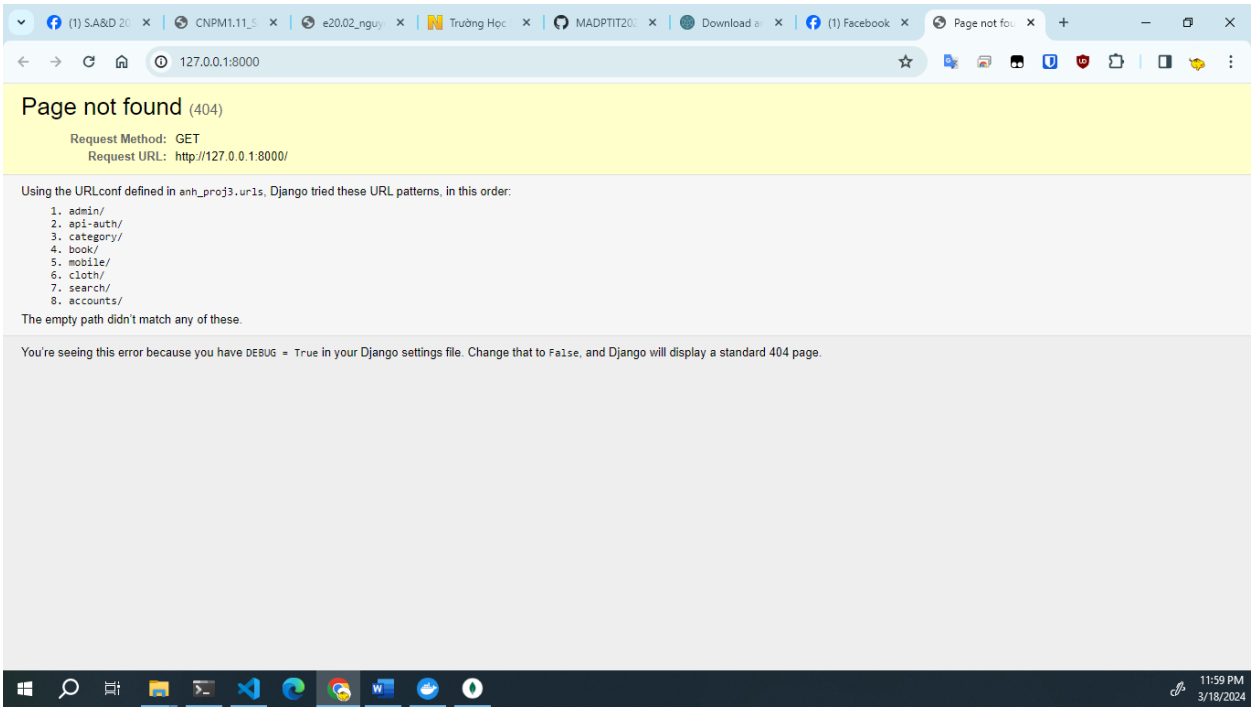


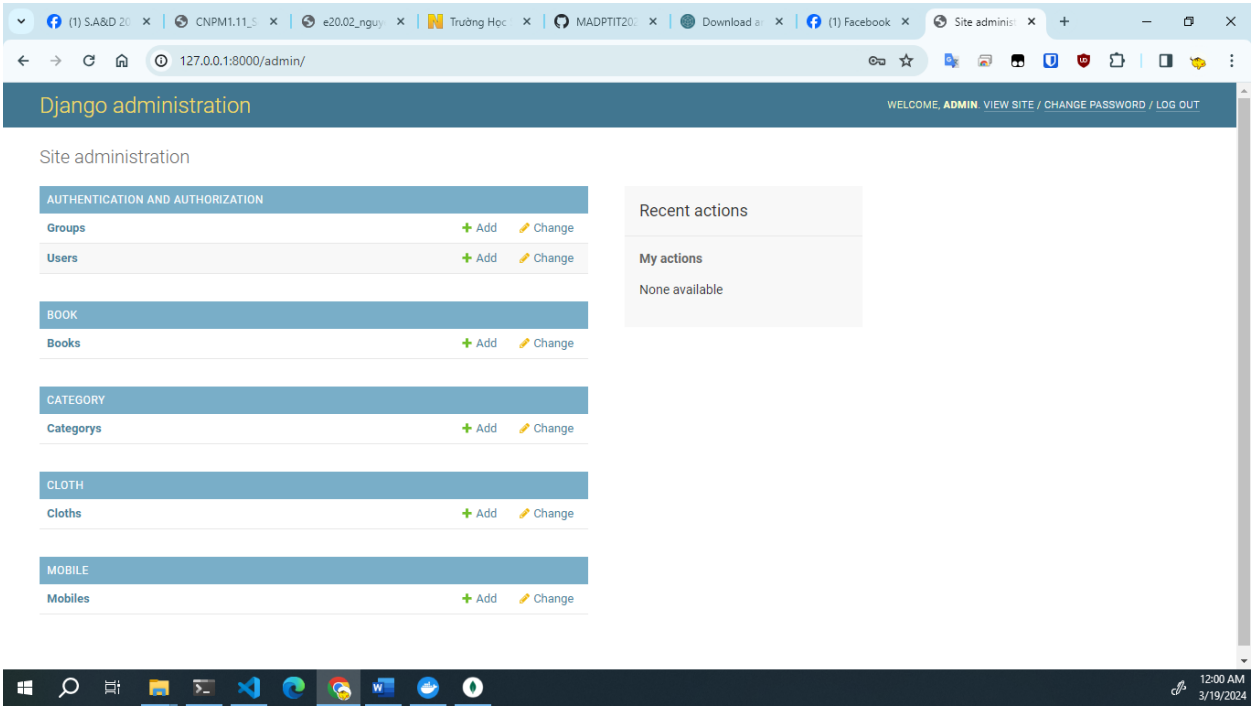


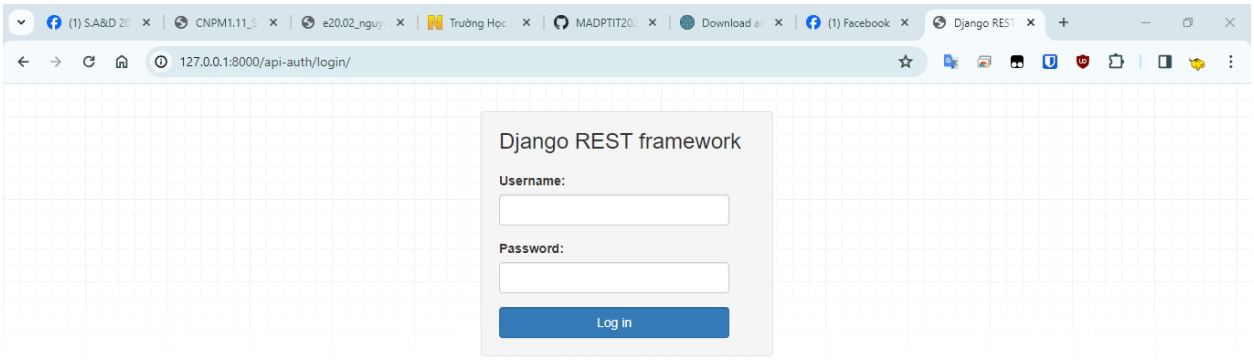
- Chạy lại project:
  Python manage.py runserver

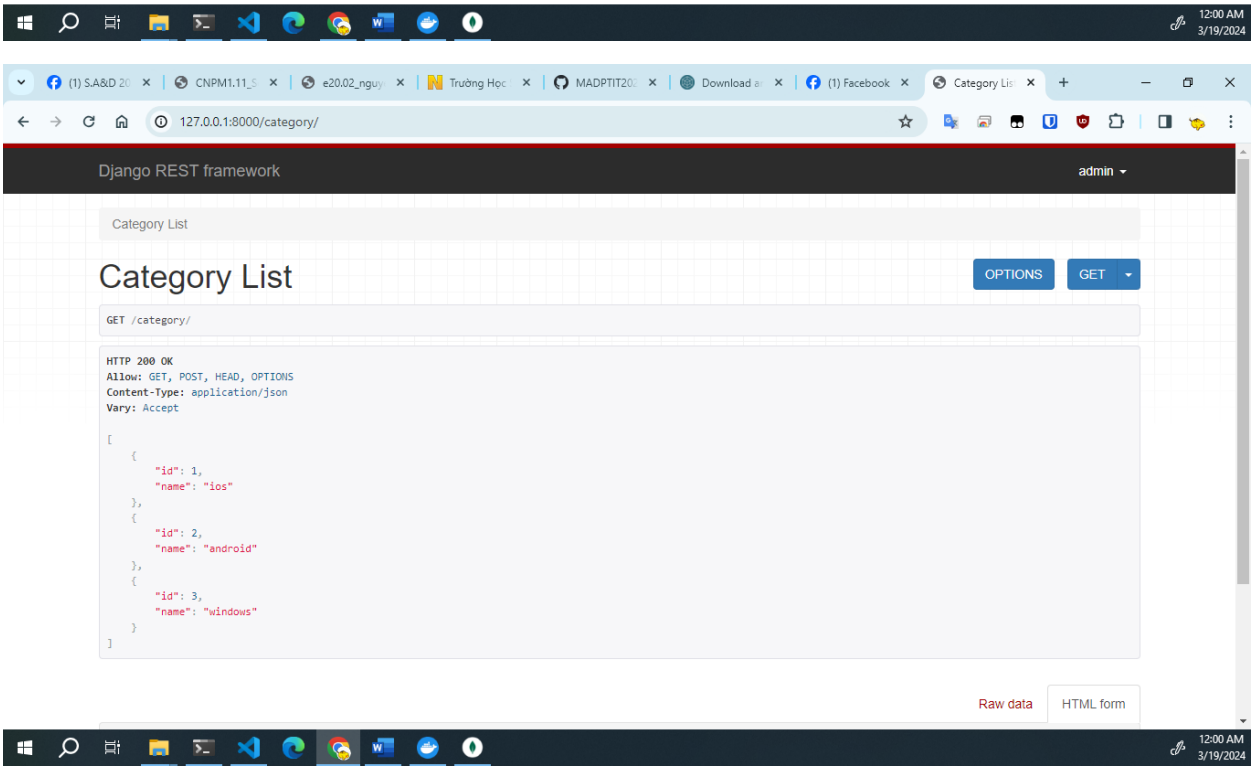


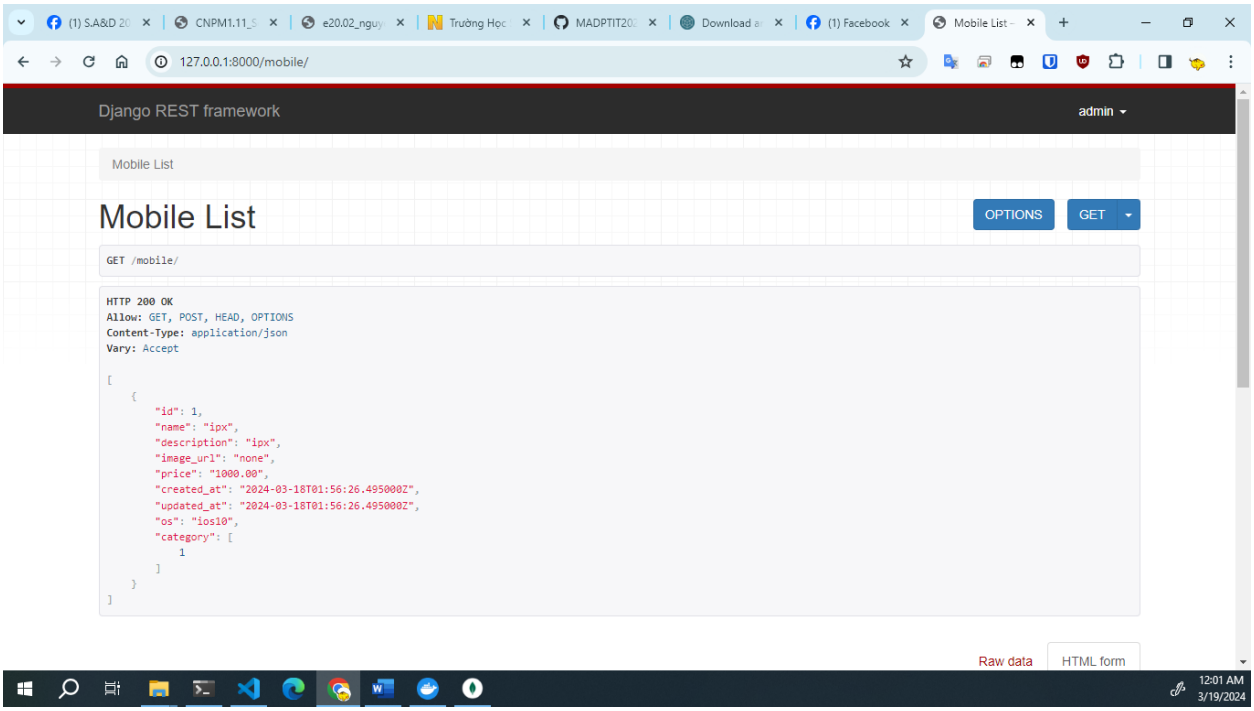
III. Testing API

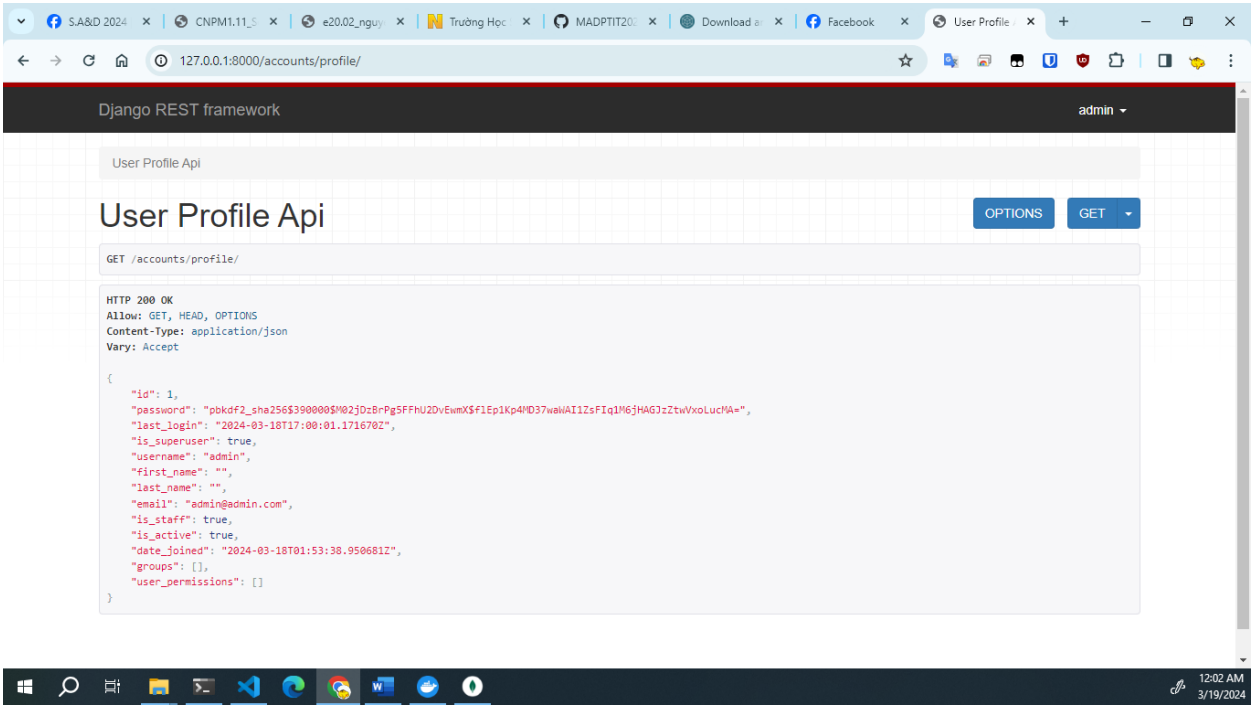

127.0.0.1:8000
Page not found (404)
Request Method: GET
Request URL: http://127.0.0.1:8000/
Using the URLconf defined in anh_proj3.urls, Django tried these URL patterns, in this order:
1. admin/
2. api-auth/
3. category/
4. book/
5. mobile/
6. cloth/
7. search/
8. accounts/
The empty path didn't match any of these.
You're seeing this error because you have DEBUG = True in your Django settings file. Change that to False, and Django will display a standard 404 page.


127.0.0.1:8000/admin/
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
Users + Add Change
BOOK
Books + Add Change
CATEGORY
Categorys + Add Change
CLOTH
Cloths + Add Change
MOBILE
Mobiles + Add Change
Recent actions
My actions
None available


127.0.0.1:8000/api-auth/login/
Django REST framework
Username:
Password:
Log in


127.0.0.1:8000/category/
Django REST framework
admin
Category List
Category List
OPTIONS
GET
GET /category/
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
[
{
"id": 1,
"name": "ios"
},
{
"id": 2,
"name": "android"
},
{
"id": 3,
"name": "windows"
}
]
Raw data
HTML form


Django REST framework
admin
Mobile List
Mobile List
OPTIONS
GET
GET /mobile/
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
[
{
"id": 1,
"name": "ipx",
"description": "ipx",
"image_url": "none",
"price": "1000.00",
"created_at": "2024-03-18T01:56:26.495000Z",
"updated_at": "2024-03-18T01:56:26.495000Z",
"os": "ios10",
"category": [
1
]
}
]
Raw data
HTML form


Django REST framework
admin
User Profile Api
User Profile Api
OPTIONS
GET
GET /accounts/profile/
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
{
"id": 1,
"password": "pbkdf2_sha256$390000$M02jDzBrPgSFFhU2DvEwmX$f1Ep1Kp4MD37waWAI1ZsFIq1M6jHAGJzZtwVxoLucMA=",
"last_login": "2024-03-18T17:00:01.171670Z",
"is_superuser": true,
"username": "admin",
"first_name": "",
"last_name": "",
"email": "admin@admin.com",
"is_staff": true,
"is_active": true,
"date_joined": "2024-03-18T01:53:38.950681Z",
"groups": [],
"user_permissions": []
}